

GRAPHIS—Visualise, Draw, Annotate, and Save Image Regions in Graffiti Photos

Geert J. Verhoeven ^{1,*}, Martin Wieser ² and Massimiliano Carloni ³

¹University of Vienna - Department of Prehistoric and Historical Archaeology, 1190 Vienna, Austria; E-Mails: geert@projectindigo.eu; geert.verhoeven@univie.ac.at

²Independent researcher, Vienna, Austria; E-Mail: scene2map@gmail.com

³Austrian Centre for Digital Humanities and Cultural Heritage, Austrian Academy of Sciences, 1010 Vienna, Austria; E-Mail: Massimiliano.Carloni@oeaw.ac.at

* Corresponding author

Abstract

A digital photo file contains the image pixel values along with associated photo metadata. Storing those metadata is enabled by various standards. For instance, the Exif standard enables the recording of technical photo metadata like the camera's serial number and focal length, while the IPTC Photo Metadata Standard is the widely accepted norm for storing copyright and descriptive information in images (from unedited photos to AI-generated pictures). Since its 2019.1 version, the IPTC Photo Metadata Standard has facilitated the creation of image regions: groupings of image pixels—defined by a circle, rectangle, or any other polygonal shape—which can be annotated with region-specific metadata. Given the potential of image regions for graffiti photo annotation, the open-source and freely available software GRAPHIS was developed within the academic graffiti project INDIGO. GRAPHIS (Generate Regions and Annotations for PHotos using the IPTC Standard) allows users to generate and visualise image regions, annotate them with graffiti descriptions or transcriptions, and save them as metadata within the image. To adhere to the IPTC Photo Metadata Standard at every stage, project INDIGO also created a dedicated controlled vocabulary to contain all relevant concepts that can be used to define each image region's role and content type. This paper starts with a general overview of metadata concepts, followed by a more in-depth look at Exif and IPTC photo metadata. After describing the IPTC *Image Region* property, the text details the workings of GRAPHIS and the controlled vocabulary development. An overview of use cases and potential software improvements conclude the text.

Keywords

Annotate; Graffiti-scape; Graffito; GRAPHIS; Image region; IPTC; Metadata; Photograph; Polygon

1. Introduction

1.1. Data and Metadata

The two photographs in Figure 1 depict graffiti created along Vienna's *Donaukanal* (Eng. Danube Canal). Each photo was acquired on the 30th of October 2023 with a Nikon Z7 II full-frame mirrorless camera paired with a Nikon NIKKOR

Z 20mm f/1.8 S lens. For both photos, the lens featured an aperture of f/5.6. However, the camera's shutter speed and ISO were 1/400 s and 320 for the left photo, and 1/1250 s and 64 for the right photo. The photo on the left was taken forty-two minutes before the photo on the right, created at 11:12:11 Central European Time (CET).



Nikon Z7 II
Nikon NIKKOR Z 20mm f/1.8 S

30 October 2023
10:30:52 CET

aperture priority
1/400 s
f/5.6
320

camera model
lens model

acquisition date
acquisition time

exposure program
exposure time
aperture
ISO

Nikon Z7 II
Nikon NIKKOR Z 20mm f/1.8 S

30 October 2023
11:12:11 CET

aperture priority
1/1250 s
f/5.6
64

Figure 1. Two graffiti photos and some of their technical metadata.

In information science jargon, the photos are called *data*, *subjects*, *potentially informative objects* or *(information) resources*, while the information describing those photos is termed *metadata* (Pomerantz, 2015). Metadata are information on the “what, when, where, how and who” of data. The ISBN or title of a book are examples of metadata, but so are the genre and rating of a movie or a photo’s exposure parameters and creation date. Metadata often unlock the value of resources, because metadata elements can describe, locate and explain the data, making them retrievable, (re-)useable, and manageable. That is why the ISO 15489-1:2016 Information and documentation standard defined metadata as “structured or semi-structured information, which enables the creation, management, and use of records through time and within and across domains” (International Organization for Standardization, 2016, p. 2). Since data is a plural noun (Borgman, 2015; Bryson, 2008), data—and thus metadata—are when referred to as entities rather than concepts.

So, metadata are statements about data or resources. These statements usually take the form of *triples* (see Figure 2); in other words, they feature three parts:

- The *subject*—or (information) resource, data, potentially informative object;
- The *predicate*—or element, property, field, attribute, characteristic;
- The *object*—or value.

For example, taking Banksy’s Flower Thrower graffiti as the starting point, it can be stated that Banksy (object) is the creator (predicate) of the Flower Thrower graffiti (subject). Note that *object* is what we call *subject* in grammar. Yes, it is somewhat confusing! In metadata schemes, the predicates are referred to as metadata elements or fields, but they can also be considered the resource’s attributes, properties or characteristics. Each metadata element has a value (here, Banksy), although some might also be left blank. Such an element-value pair is a single statement about a resource

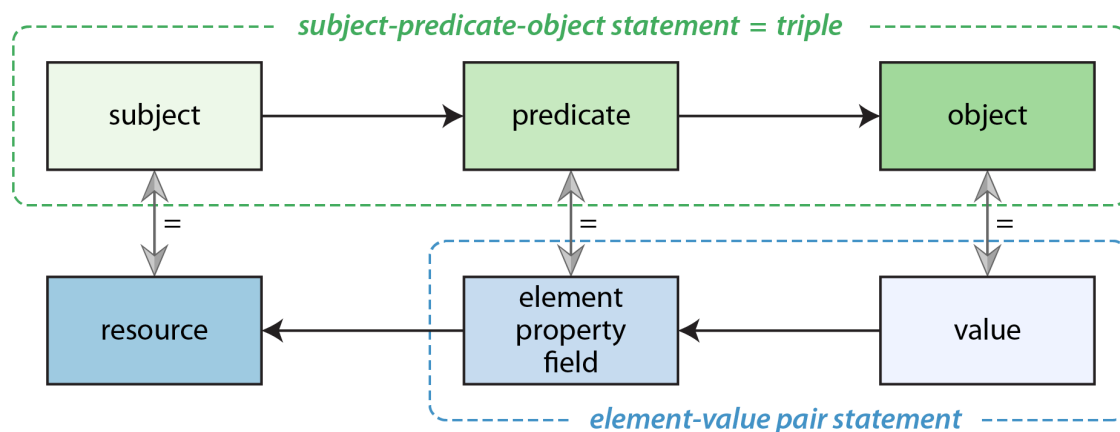


Figure 2. The basic elements of metadata.

(see Figure 2), while a set of statements is a metadata record (Pomerantz, 2015).

Many different element-value pairs constitute all the statements (or the metadata record) about the resource. A *metadata schema* defines which elements are allowed, their optionality, how many separate values they can have, as well as any potential parent/child relationships between these elements. Most metadata schemas include an *application profile* that delineates how and when to use the metadata elements, while the rules for selecting or constructing these elements' values are specified in the metadata *encoding scheme* (although different communities put different emphasis on these two terms; ISO/TC 46/SC11, 2008). The encoding scheme consists of a *syntax encoding* to stipulate how specific values must be represented (for example, data elements should follow the ISO 8601 encoding scheme) and *controlled vocabularies* to provide finite lists of values for specific elements. So overall, a metadata schema is a formal description for data, containing a set of rules about the subject-predicate-object statements that can be made (Pomerantz, 2015).

Since various general and domain-specific metadata schemas have been developed over the past decades, it is seldom necessary to create a new one, and it is typically better to adopt (and maybe adapt) a proven, well-supported

metadata schema. One or more of these schemas often form accepted *metadata standards* like Dublin Core (<https://www.dublincore.org>), IPTC Photo Metadata Standard (<https://iptc.org/standards/photo-metadata/iptc-standard>) or Darwin Core (<https://dwc.tdwg.org>).

1.2. Encoding Metadata Schemas: RDF and XML

Metadata are partly founded on structured data, which are data organised or structured according to a data model. A data model is a framework, a logical structure to represent all the resource types contained by the data, the properties of those resources, and the relationships between them (Pomerantz, 2015). Many data models exist, but the *Resource Description Framework* (RDF; <https://www.w3.org/RDF>) developed by the World Wide Web Consortium (W3C) is the data model that structures most metadata. In other words, RDF is a generic data model or logical structure that defines how resources should be described.

RDF is based on the triples or subject-predicate-object statements introduced in Section 1.1. For example (see Figure 3), the statement “indigo is the colour of that mural” can be broken down into:

- “Mural” or the resource being described, known as the *subject*;
- “Indigo” or the value of that description, known as the *object*;

- “Colour”, which establishes the relationship between the *subject* and *object*, known as the *predicate*. Sometimes, the predicate would also be written as “hasColour”.

When additional metadata statements about that mural (e.g., “2023-06-15 is the creation date of that mural” and “Cope26 is the creator of that mural”) complement the above statement, one ends up with a set of RDF triples. Such a set of entities connected by relationships is known as a *graph* (see Figure 3). Now imagine that Mr. Wogrin would photograph that mural, resulting in a digital photo called muralPhoto.jpg. At that moment, there would be two resources—a physical one (i.e., the mural, a real-world graffiti) and its digital approximation (i.e., the muralPhoto.jpg file, a digital photograph)—each featuring the *Creator* or *hasCreator* predicate (see Figure 3). Since this distinction

between physical and digital resources is important but often ignored, Section 1.3 will further explore it.

The specific syntax used for encoding the RDF triples in a metadata schema is typically XML-based. XML, or the eXtensible Markup Language, describes a set of rules for structuring documents in a human- and machine-readable format (Adobe Systems Incorporated, 2012). XML contains instructions or *tags* (enclosed in angle brackets) that can be incorporated into text documents, but these tags can also structure arbitrary data. The tags (which have nothing to do with the eponymous graffiti type) are placed around some content to make up an *element*. Consider again the assertion “Cope26 is the creator of that mural”, which can be represented with some simplifications in XML as:

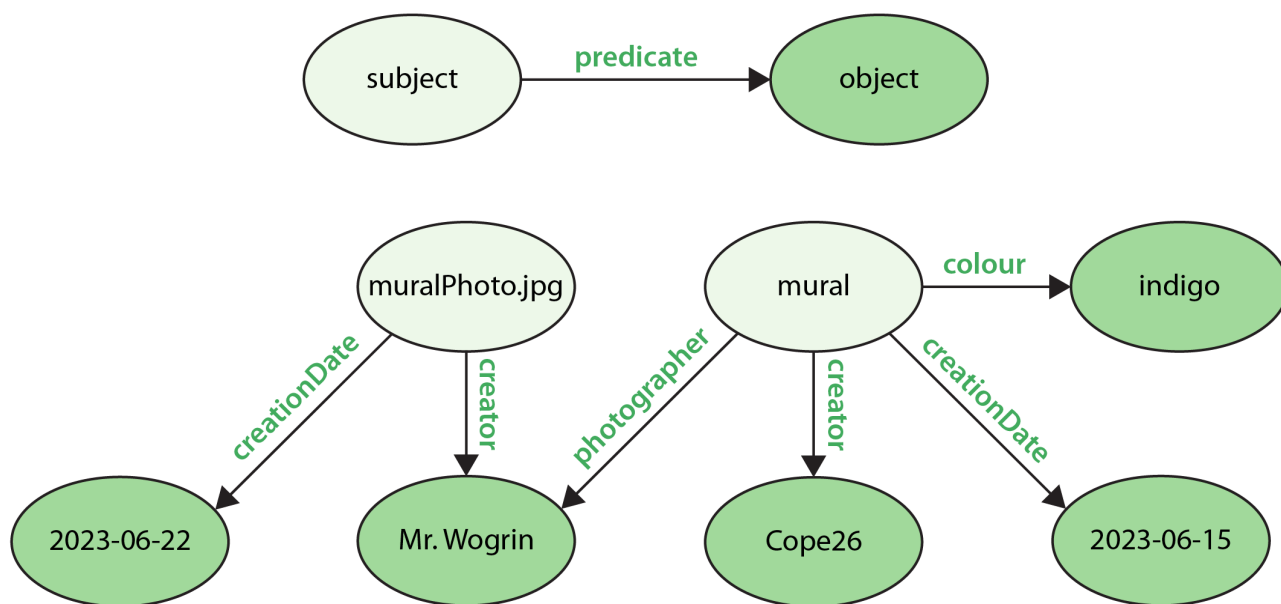


Figure 3. Top: RDF structures metadata in subject-predicate-object triples. Bottom: many RDF triples may link up to form a graph.

```
<mural>
  <creator>
    <name>Cope26</name>
  </creator>
</mural>
```

The element <creator> consists of a sub-element <name>. This sub-element has some text as content. Note that a slash character precedes the closing tag and that the hierarchy among the elements lends itself to a tree representation. XML documents are always formed as *element trees*. Every

XML tree starts at a root element and branches to sub-elements, which themselves can further branch to sub-elements. However, the XML tags one can use are not predefined. A specific tagset can be defined for each use case, making this markup language *extensible*.

By encoding RDF graphs via an XML document, RDF information is easily exchangeable between different computers using differing operating systems and applications. The XML syntax used for RDF is known as RDF/XML, a standard developed by the W3C and available at <https://www.w3.org/TR/rdf-syntax-grammar>. RDF/XML lies at the core of the Semantic Web (Yu, 2014) as well as Adobe's XMP technology, which is leveraged by the IPTC Photo Metadata Standard (see Section 2.4).

1.3. Resource-Specific Metadata

Much like how “A map is not the territory it represents, but, if correct, it has a similar structure to the territory, which accounts for its usefulness” (Korzybski, 1933, p. 58), a photograph of a physical, real-world graffito is not the graffito, but an approximation which can be analogue or digital. Similar to how different maps represent different characteristics of the physical world (e.g. road maps, nautical maps, topographical maps), different approximations like photos, sketches, and digital 3D surface models can be used to boil down all of a physical graffito's complexity to those aspects that are needed for a particular purpose or in a specific situation.

In the academic graffiti project INDIGO (<https://projectindigo.eu>), all graffiti approximations were digital, and none of these digital approximations equal the analogue physical graffito. Still, they allow us to obtain information about it. Although the original graffito and its digital approximations are both referred to as data or resources, they are separate entities that need their specific metadata in terms of the metadata elements and their values (see Figure 4). For example, “camera model” is an irrelevant metadata element for the real graffito (which is a physical resource), but valuable information about the digital graffito photo (which is an electronic or digital resource); “creator” and “copyright” are elements relevant for both but populated with different, resource-dependent values.

Creating metadata for cultural heritage assets often ignores this distinction between original and derived resources, even though the guidelines formulated by the Visual Resources Association for cataloguing cultural objects have already advocated this in 2006 (Baca et al., 2006). INDIGO made the distinction between these resources explicit and created a metadata schema for physical graffiti that differs from the metadata attached to the digital (ortho)photos or textured 3D surface models of those graffiti (see Figure 4). Since there should always be exactly one metadata record for a single resource—known as the *one-to-one principle* in information science (Pomerantz, 2015)—one metadata record per graffito, per graffito photo, per graffito orthophoto, per graffito 3D model was the correct way to proceed.

In addition, distinguishing between the physical graffito and its digital approximations also helps improve metadata provenance tracking. For example, much of the physical graffito's metadata cannot, or not easily, be observed *in situ*. Good examples are a large graffito's maximum dimensions or total area, which are much easier to derive via a digital 3D model. Afterwards, one can transfer that information to the metadata of the physical graffito (see Figure 4). This train of thought guided the development of GRAPHIS, a software tool to trace and annotate a graffito's border and store that information as a so-called image region in the photo's metadata. This image region is useful for a multitude of purposes, including calculating the dimensions and area of that graffito.

The remainder of this paper will first provide an overview of the two prevalent metadata standards for digital photographs (Section 2) and focus on the *Image Region* metadata element introduced in 2019 by one of them (Section 3). Section 4 then details the inner workings of the GRAPHIS tool built to create, visualise, and manage this *Image Region* element, as well as the controlled vocabulary used to limit some of the region's attributes. Before concluding the paper, Section 5 delves deeper into the multiple purposes this image region could serve.

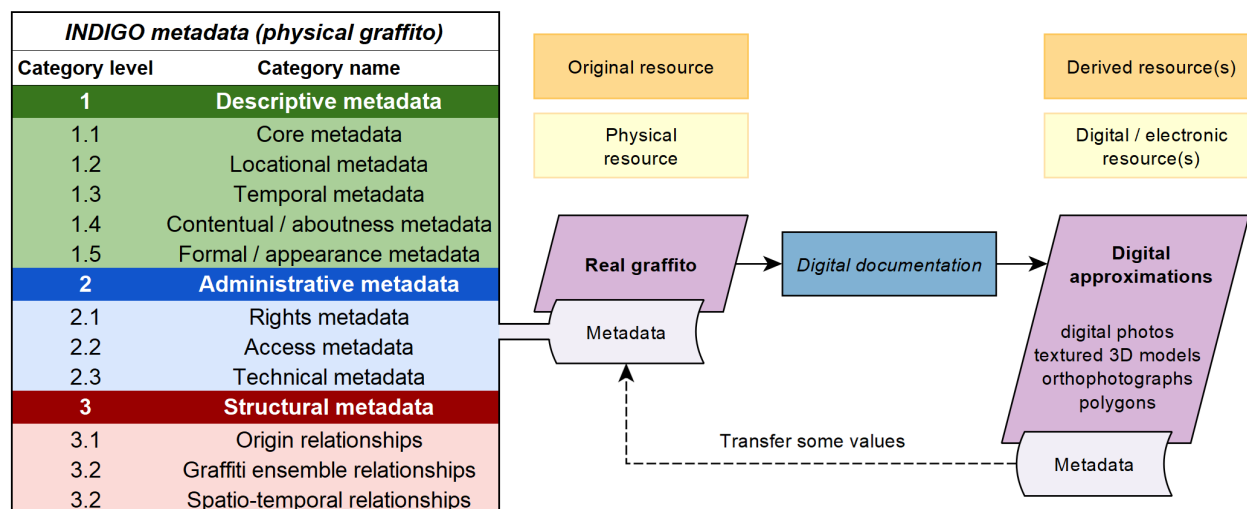


Figure 4. Various ways exist to categorise or group metadata. To describe a physical graffiti, INDIGO's metadata schema groups information into descriptive, administrative, and structural sections, a grouping that follows Gartner (2016) and Horodyski (2022). Values for some descriptive metadata fields of a real graffiti (like “surface area” or “colours used”) are sourced from the metadata of its digital approximations. Besides some information on the descriptive and administrative metadata of digital graffiti photos—implemented via the Exif and IPTC Photo Metadata Standards—the details of INDIGO's metadata schema for each type of digital approximation are beyond the scope of this paper.

2. Photo Metadata

2.1. Flavours of Pixel Values

Photographs created by a digital camera typically come in two main types: photos containing raw pixel values and fully processed photos where every pixel has a Red-Green-Blue (RGB) colour value. This dichotomy or ‘choice’ is also reflected in many digital cameras, which can usually produce and store both image types:

1. An image with one digital value per pixel, corresponding to the amount of photons captured by the camera's imaging sensor for that location. This image is typically referred to as the RAW photo or RAW file. RAW is not an acronym nor a file extension. It only signifies raw or minimally processed image sensor data with pixel values that are linearly related to the incoming radiation in the Red, Green or Blue visible spectral band. RAW can thus be considered the only scientifically justifiable file format (Verhoeven, 2010). However, RAW files come with manufacturer-specific structures and extensions (like .nef for Nikon, .raf for Fuji, and .crw or .cr2 for Canon RAW photos), and the raw data need many

processing steps to produce the second image type: a normal-looking photo.

2. A highly processed viewable image with pixels nonlinearly related to the captured amount of photons. This image is usually expressed in the sRGB colour space and saved as a *.jpg/*.jpeg or *.tiff file. This viewable type of image is commonly meant when talking about a photo. Even though some dedicated cameras (and smartphones) might not offer the option to save the RAW image, the latter always internally forms the basis to yield a viewable output photo.

A digital photo can thus store two ‘types’ of primary image data; typically, one of these two types is stored, although some RAW files might also contain a viewable output photo. Besides those data, the photo file might include one or more thumbnails. Together, the primary image data and the thumbnail(s) constitute all the pixel values of a digital photo file (see Figure 5). However, that photo file can also serve as a container for metadata.

2.2. Metadata Containers

Metadata state something about an analogue or physical resource, but those metadata must also exist in a physical or digital container. In other words, metadata can be located inside or outside the resource, and the location often depends on the resource and its use case. Examples of internal metadata are the copyright and title of a book on its copyright page. However, one could also save this info in a library catalogue card box as external metadata. For a digital photo (see Figure 5), the metadata record is either:

- Embedded in the photo file, therefore known as *internal metadata*. The Exif information (see Section 2.3) or IPTC-IIM values (see Section 2.4) stored in the header of a digital photograph are prime examples of this. Upon acquisition, RAW and JPEG or TIFF photos have their Exif metadata automatically stored within the file by the digital camera;

- Stored externally, either in:
 - o A separate but related file (like an *.xmp sidecar file for RAW digital photographs; see Section 5.2);
 - o A database like a Digital Asset Management (DAM) or Media Asset Management (MAM) system that links to the photo file via a unique identifier. Besides a DAM or MAM, metadata can be stored in a data repository (like ARCHE; see the paper by Trognitz et al. in this volume).

Some metadata fields must be populated manually, while software applications or firmware auto-generate the values of others. IPTC metadata—detailed in Section 2.4—are an example of the former, while Exif metadata (described in the following section) are typically auto-generated within the digital camera.

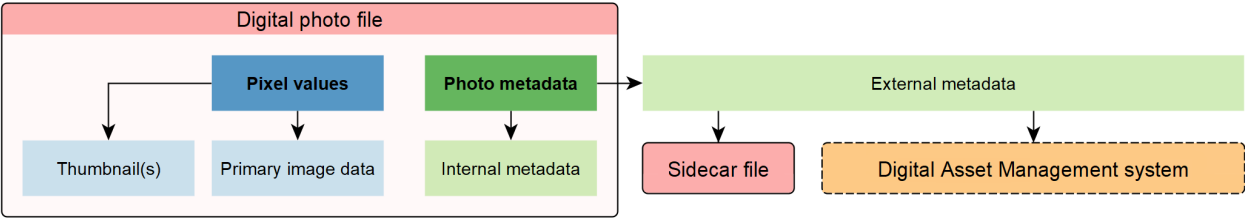


Figure 5. Photo metadata are either stored internally beside the pixel values or externally in a sidecar file or database.

2.3. Exif Metadata

In addition to the pixel values that encode the real-world scene, RAW files and JPEG or TIFF photos contain Exif or *Exchangeable image file format* metadata. These technical metadata describe image acquisition parameters—such as the serial number and model of the camera, the aperture, focal length, shutter speed, possible flash compensation, and the date plus time of photo acquisition—in mandatory, recommended, and optional fields (called *tags*) stored in a separate segment of the photo file (Camera & Imaging Products Association, 2023). Suppose the camera is GNSS (Global Navigation Satellite System)-enabled. In that case, tags can also hold the latitude, longitude, and altitude of the camera’s geographical location. All these Exif-defined tags are created by the camera and stored simultaneously

with the pixel values in the image file, making it possible to analyse them afterwards.

The first Exif standard (version 1.0) was released in October 1995 by JEIDA, the Japan Electronic Industries Development Association (see Figure 6). JEIDA also published versions 1.1 (May 1997), 2.0 (November 1997), and 2.1 (June and December 1998). Because JEIDA became JEITA (Japan Electronics and Information Technology Industries Association) in November 2000 (JEITA, 2000), Exif version 2.2 (April 2002) was established by JEITA. Since 2009, JEITA has been launching the Exif standard jointly with CIPA, the Camera & Imaging Products Association. CIPA and JEITA co-published Exif version 3.0 in May 2023.

2.4. IPTC Metadata

Whereas Exif metadata mainly contain technical information about the digital photo creation process, the IPTC Photo Metadata Standard is widely accepted for storing non-technical information about photos (or, more generally, images). Professional photographers, news agencies,

museums, and libraries rely on the IPTC metadata properties to describe the image content (i.e., descriptive metadata), provide instructions for the users (i.e., access metadata) or store its copyright information (i.e., rights metadata). These descriptive, access, and rights-related properties are stored within or along the image file. Ideally, all tools that read

Exif metadata specification history

Version	Standard Number	Issue Date	Published by
1.0		1995 10	JEIDA
1.1		1997 05	JEIDA
2.0		1997 11	JEIDA
2.1		1998 06	JEIDA
2.1 (DCF version)	JEIDA-49-1998	1998 12	JEIDA
2.2	JEITA CP-3451	2002 04	JEITA
2.21	JEITA CP-3451-1	2003 09	JEITA
Unified version 2.21	CIPA DC-008-2009 / JEITA CP-3451A	2009 09 01	CIPA & JEITA
2.3	CIPA DC-008-2010 / JEITA CP-3451B	2010 04 26	CIPA & JEITA
2.3 (Revised 2012)	CIPA DC-008-2012 / JEITA CP-3451C	2012 12	CIPA & JEITA
2.31	CIPA DC-0008-2016 / JEITA CP-3451D	2016 07	CIPA & JEITA
2.32	CIPA DC-0008-2019 / JEITA CP-3451E	2019 05 17	CIPA & JEITA
3.0	CIPA DC-008-2023 / JEITA CP-3451F	2023 05 29	CIPA & JEITA

Figure 6. The history of the Exif metadata specification.

and write IPTC image metadata should keep all metadata embedded into the image file (for *.jpg, *.tiff or *.png files) or saved as an identically named sidecar *.xmp file (for RAW files) persistent when exchanged between various software and users.

As in every metadata standard, each IPTC Photo Metadata field is tightly defined. The latest version of the IPTC Photo Metadata Standard (i.e., version 2023.2; <https://iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-2023.2.html>) defines 62 top metadata fields (IPTC Photo Metadata Working Group, 2024) (see Figure 7). Many of those 62 are *single fields* that store one value to express the desired information, but some are *field structures* that contain multiple sub-fields. In Figure 7, field structures have *struct* as their data type. The IPTC uses the generic term *property* for a *field structure* or a *single field* (commonly shortened to *field*).

Those 62 properties are divided into two metadata schemas: the IPTC Core and the IPTC Extension schema, each with a specific development history and version (see Figure 8). The IPTC Photo Metadata Standard 2023.2 contains the

IPTC Core schema 1.4 with 25 top properties and the IPTC Extension schema 1.8 with 37 top properties (see Figure 7).

Initially, the *IPTC Core schema* started in 2004 as a revamp of the *Information Interchange Model* (IIM; <https://www.iptc.org/standards/iim>) standard by the IPTC and the American Newspaper Publishers Association (ANPA) (see Figure 9 on the left). The first version of the IPTC IIM multimedia standard was launched in 1990. Since ANPA became the Newspaper Association of America (NAA) in 1992, the IPTC IIM standard also became known as the IPTC-NAA IIM standard.

The IIM defined a series of metadata fields such as *Object Name*, *Edit Status*, *Urgency*, *Date Created*, and *Keywords* to aid the interchange of news between computerised systems. Even though those metadata fields were media-type agnostic (i.e., useable for text, video, audio, and photos), they could be embedded inside digital images thanks to the *image resource block* technology developed by Adobe Systems Incorporated (Adobe Systems Incorporated, 1991-2008).

Property	Schema	Data type	Data format	Occurrence	Required	Part of IIM	XMP identifier
Alt Text (Accessibility)	IPTC Core	struct	Alt.Lang	single	☐	☐	Iptc4xmpCore:AltTextAccessibility
City (legacy)	IPTC Core	string		single	☐	☒	photoshop:City
Copyright Notice	IPTC Core	struct	Alt.Lang	single	☐	☒	dc:rights
Country (legacy)	IPTC Core	string		single	☐	☒	photoshop:Country
Country Code (legacy)	IPTC Core	string		single	☐	☒	Iptc4xmpCore:CountryCode
Creator	IPTC Core	string		multi	☐	☒	dc:creator
Creator's Contact Info	IPTC Core	struct	CreatorContactInfo	single	☐	☒	Iptc4xmpCore:CreatorContactInfo
Creator's jobtitle	IPTC Core	string		single	☐	☒	photoshop:AuthorsPosition
Credit Line	IPTC Core	string		single	☐	☒	photoshop:Credit
Date Created	IPTC Core	string	date-time	single	☐	☒	photoshop:DateCreated
Description	IPTC Core	struct	Alt.Lang	single	☐	☒	dc:description
Description Writer	IPTC Core	string		single	☐	☒	photoshop:CaptionWriter
Extended Description (Accessibility)	IPTC Core	struct	Alt.Lang	single	☐	☐	Iptc4xmpCore:ExtDescrAccessibility
Headline	IPTC Core	string		single	☐	☒	photoshop:Headline
Instructions	IPTC Core	string		single	☐	☒	photoshop:Instructions
Intellectual Genre (legacy)	IPTC Core	string		single	☐	☒	Iptc4xmpCore:IntellectualGenre
Job Id	IPTC Core	string		single	☐	☒	photoshop:TransmissionReference
Keywords	IPTC Core	string		multi	☐	☒	dc:subject
Province or State (legacy)	IPTC Core	string		single	☐	☒	photoshop:State
Rights Usage Terms	IPTC Core	struct	Alt.Lang	single	☐	☐	xmpRights:UsageTerms
Scene Code	IPTC Core	string		multi	☐	☒	Iptc4xmpCore:Scene
Source (Supply Chain)	IPTC Core	string		single	☐	☒	photoshop:Source
Subject Code (legacy)	IPTC Core	string		multi	☐	☒	Iptc4xmpCore:SubjectCode
Sublocation (legacy)	IPTC Core	string		single	☐	☒	Iptc4xmpCore:Location
Title	IPTC Core	struct	Alt.Lang	single	☐	☒	dc:title
Additional Model Information	IPTC Extension	string		single	☐	☐	Iptc4xmpExt:AddlModelInfo
Artwork or Object in the Image	IPTC Extension	struct	ArtworkOrObject	multi	☐	☐	Iptc4xmpExt:ArtworkOrObject
Code of Organisation Featured in the Image	IPTC Extension	string		multi	☐	☐	Iptc4xmpExt:OrganisationInImageCode
Contributor	IPTC Extension	struct	EntityWRole	multi	☐	☐	Iptc4xmpExt:Contributor
Copyright Owner	IPTC Extension	struct	CopyrightOwner	multi	☐	☐	plus:CopyrightOwner
CV-Term About Image	IPTC Extension	struct	CvTerm	multi	☐	☐	Iptc4xmpExt:AboutCvTerm
Data Mining	IPTC Extension	string	uri	single	☐	☐	plus:DataMining
Other Constraints	IPTC Extension	struct	Alt.Lang	single	☐	☐	plus:OtherConstraints
Digital Image GUID	IPTC Extension	string		single	☐	☐	Iptc4xmpExt:DigImageGUID
Digital Source Type	IPTC Extension	string	uri	single	☐	☐	Iptc4xmpExt:DigitalSourceType
Embedded Encoded Rights Expression	IPTC Extension	struct	EmbdEncRightsExpr	multi	☐	☐	Iptc4xmpExt:EmbdEncRightsExpr
Event Identifier	IPTC Extension	string	uri	multi	☐	☐	Iptc4xmpExt:EventId
Event Name	IPTC Extension	struct	Alt.Lang	single	☐	☐	Iptc4xmpExt:Event
Genre	IPTC Extension	struct	CvTerm	multi	☐	☐	Iptc4xmpExt:Genre
Image Creator	IPTC Extension	struct	ImageCreator	multi	☐	☐	plus:ImageCreator
Image Rating	IPTC Extension	number		single	☐	☐	xmp:Rating
Image Region	IPTC Extension	struct	ImageRegion	multi	☐	☐	Iptc4xmpExt:ImageRegion
Image Registry Entry	IPTC Extension	struct	RegistryEntry	multi	☐	☐	Iptc4xmpExt:RegistryId
Image Supplier	IPTC Extension	struct	ImageSupplier	multi	☐	☐	plus:ImageSupplier
Image Supplier Image ID	IPTC Extension	string		single	☐	☐	plus:ImageSupplierImageId
Licensor	IPTC Extension	struct	Licensor	multi	☐	☐	plus:Licensor
Linked Encoded Rights Expression	IPTC Extension	struct	LinkedEncRightsExpr	multi	☐	☐	Iptc4xmpExt:LinkedEncRightsExpr
Location created	IPTC Extension	struct	Location	single	☐	☐	Iptc4xmpExt:LocationCreated
Location Shown in the Image	IPTC Extension	struct	Location	multi	☐	☐	Iptc4xmpExt:LocationShown
Max Avail Height	IPTC Extension	number	integer	single	☐	☐	Iptc4xmpExt:MaxAvailHeight
Max Avail Width	IPTC Extension	number	integer	single	☐	☐	Iptc4xmpExt:MaxAvailWidth
Minor Model Age Disclosure	IPTC Extension	string	uri	single	☐	☐	plus:MinorModelAgeDisclosure
Model Age	IPTC Extension	number	integer	multi	☐	☐	Iptc4xmpExt:ModelAge
Model Release Id	IPTC Extension	string		multi	☐	☐	plus:ModelReleaseId
Model Release Status	IPTC Extension	string	uri	single	☐	☐	plus:ModelReleaseStatus
Name of Organisation Featured in the Image	IPTC Extension	string		multi	☐	☐	Iptc4xmpExt:OrganisationInImageName
Person Shown in the Image	IPTC Extension	string		multi	☐	☐	Iptc4xmpExt:PersonInImage
Person Shown in the Image with Details	IPTC Extension	struct	PersonWDetails	multi	☐	☐	Iptc4xmpExt:PersonInImageWDetails
Product Shown in the Image	IPTC Extension	struct	ProductWGTin	multi	☐	☐	Iptc4xmpExt:ProductInImage
Property Release Id	IPTC Extension	string		multi	☐	☐	plus:PropertyReleaseId
Property Release Status	IPTC Extension	string	uri	single	☐	☐	plus:PropertyReleaseStatus
Web Statement of Rights	IPTC Extension	string	uri	single	☐	☐	xmpRights:WebStatement

Figure 7. The 62 top metadata properties of the IPTC Photo Metadata Standard 2023.2 are divided into 37 single fields and 25 field structures. All information originates from the IPTC Photo Metadata Technical Reference Documentation (https://iptc.org/std/photometadata/specification/iptc-pmd-techreference_2023.2.json).

IPTC Core schema specification history					IPTC Extension schema specification history				
Version	Revision	Approval date	Approved by	In Standard Version	Version	Revision	Approval date	Approved by	In Standard Version
1.0		2004 10 08	IPTC Standards Committee	2004	1.0		2008 07 02	IPTC Standards Committee	2008
1.0	1	2004 10 10	IPTC Standards Committee	2004	1.1		2009 06 17	IPTC Standards Committee	July 2009
1.0	2	2004 10 14	IPTC Standards Committee	2004	1.2		2014 10 22	IPTC Standards Committee	July 2014
1.0	3	2004 11 09	IPTC Standards Committee	2004	1.3		2016 10 26	IPTC Standards Committee	October 2016
1.0	4	2004 12 23	IPTC Standards Committee	2004	1.4		2017 05 17	IPTC Standards Committee	2017.1
1.0	5	2005 01 11	IPTC Standards Committee	2004	1.5		2019 10 16	IPTC Standards Committee	2019.1
1.0	6	2005 01 27	IPTC Standards Committee	2004	1.6		2021 10 20	IPTC Standards Committee	2021.1
1.0	7	2005 02 01	IPTC Standards Committee	2004	1.7		2022 10 19	IPTC Standards Committee	2022.1
1.0	8	2005 03 15	IPTC Standards Committee	2004	1.8		2023 10 04	IPTC Standards Committee	2023.1
1.1		2008 07 02	IPTC Standards Committee	2008					
1.2		2014 06 18	IPTC Standards Committee	July 2014					
1.3		2021 10 20	IPTC Standards Committee	2021.1					
1.4		2022 10 19	IPTC Standards Committee	2022.1					

Figure 8. The history of the IPTC Core and Extension schemas.

Although image resource blocks were introduced in 1992 with Adobe's Photoshop 2.5 (Murray & VanRyper, 1994), embedding IPTC-NAA IIM metadata in *.jpg, *.tiff, and *.psd image files only became possible in 1994 with the release of Photoshop 3 (Adobe Systems Incorporated & Knoll, 1990-2019). Since then, IIM metadata embeddings have been referred to as *IPTC file headers*. Due to the development of new data representation standards (like XML) in the mid-1990s, further enhancements to the IPTC-NAA IIM came to a halt in 1999 (Comité International des Télécommunications de Presse & Newspaper Association of America, 1999)—apart from a minor revision of the 1999 IIM standard in 2014 (International Press Telecommunications Council & Newspaper Association of America, 2014) (see Figure 9 on the left).

In the spring of 2004, the IPTC initiated the creation of a photograph-only metadata standard, primarily focusing on its usefulness for photographers, metadata editors, and digital image processing software. The first round of this “IPTC for XMP” or “IPTC4XMP” initiative tried to marry a set of widely used IPTC-NAA IIM metadata properties to Adobe's XMP framework (personal communication with Michael W. Steidl). XMP (or the eXtensible Metadata Platform) was introduced in 2001 by Adobe Systems Incorporated as an RDF/XML-based metadata embedding technology (Adobe Systems Incorporated, 2001a, 2001b). Using XMP, metadata could now be embedded into digital files (images and others) in an inherently extensible manner. [Note that in early 2001, Adobe initially called this metadata initiative XAP or eXtensible Authoring and Publishing (Adobe Systems Incorporated, 2002; Chapman & Brailsford, 2001). The appearance of the strings “XAP”, “xap” (e.g., <http://ns.adobe.com/xap/1.0>) or “authoring and publishing” (e.g., Adobe Systems Incorporated, 2001b) in many of the first XMP-related documents or URIs (Uniform Resource Locators) reflects this].

In the world of XMP, specific metadata properties are grouped into *namespaces*. For example, Adobe's basic XMP namespace (<https://developer.adobe.com/xmp/docs/XMPNamespaces/xmp>) contains properties that provide primary descriptive information (such as *Identifier*,

CreatorTool, *CreateDate*, and *Rating*), and the Photoshop namespace (<https://developer.adobe.com/xmp/docs/XMPNamespaces/photoshop>) specifies metadata elements used by Adobe Photoshop. Adobe also introduced the XMP Rights Management namespace (<https://developer.adobe.com/xmp/docs/XMPNamespaces/xmpRights>) for metadata properties on legal restrictions, a Camera Raw namespace (<https://developer.adobe.com/xmp/docs/XMPNamespaces/crs>) for development settings associated with RAW photos and even an Exif namespace (<https://developer.adobe.com/xmp/docs/XMPNamespaces/exif>) for specific properties typically stored in the native Exif metadata. Each namespace has a specific name, a URI, and a preferred prefix (see Figure 11). XMP properties are then commonly written in a *prefix:property* style, for example, *lptc4xmpCore:Location* (see Figures 7 and 10).

In the years before the IPTC started to work on a specific photo metadata standard in 2004, Adobe had already mapped nineteen IPTC-NAA IIM properties to the Photoshop and Dublin Core XMP namespaces for use in Photoshop's “File Info” panel (Riecks, 2005). For compatibility, the IPTC retained those mappings (personal communication with David Riecks). However, the IPTC4XMP working group also introduced in their IPTC Core namespace a few IIM properties previously not used by Adobe, as well as a handful of newly defined properties. After various revisions (see Figure 8 on the left), the first version of the XMP-based IPTC Core schema—covering a subset of the IPTC-NAA IIM properties with a few new ones—was finalised in early 2005 (International Press Telecommunications Council, 2005). However, it took three more years and the release of an IPTC Extension schema (see Figure 8 on the right) before the first IPTC Photo Metadata Standard saw the light of day in 2008 (International Press Telecommunications Council, 2008) (see Figure 9 on the right).

The IPTC Core schema currently combines properties from the Dublin Core, Photoshop, XMP Rights Management, and IPTC Core namespaces. In the column “XMP identifier” of Figure 7, these four XMP namespaces are indicated by the prefixes *dc*, *photoshop*, *xmpRights* and *lptc4xmpCore*, respectively. Although the IPTC Core metadata schema

IPTC-NAA IIM specification history			
Version	Revision	Issue Date	Approved by
1		1990 06	IPTC; American Newspaper Publishers Association (ANPA)
2		1993 04 14	
3		1995 10	IPTC; Newspaper Association of America (NAA)
4	1	1999 07	
4	2	2014 07	IPTC; Newspaper Association of America (NAA)

IPTC Photo Metadata Standard specification history					
Version	Revision	Issue Date	Approved by	IPTC Core	IPTC Extension
2008	1	2008 07 14	IPTC Standards Committee	1.1	1.0
2008	2	2008 07 18		1.1	1.0
July 2009	1	2009 07 07	IPTC Standards Committee	1.1	1.1
July 2010	1	2010 07 08		1.1	1.1
July 2014	1	2014 06 27	IPTC Standards Committee	1.2	1.1
October 2014	1	2014 11 25		1.2	1.2
October 2014	2	2014 11 28	IPTC Standards Committee	1.2	1.2
October 2014	3	2015 01 23		1.2	1.2
October 2016	1	2016 11 15	IPTC Standards Committee	1.2	1.3
2017.1	1	2017 06 19		1.2	1.4
2017.1	2	2018 06 11	IPTC Standards Committee	1.2	1.4
2017.1	3	2019 03 18		1.2	1.4
2019.1	1c	2020 01 24	IPTC Standards Committee	1.2	1.5
2021.1	1	2021 10 21		1.3	1.6
2022.1	0	2022 11 04	IPTC Standards Committee	1.4	1.7
2022.1	1	2023 01 16		1.4	1.7
2022.1	2	2023 03 01	IPTC Standards Committee	1.4	1.7
2023.1	0	2023 10 04		1.4	1.8
2023.2	0	2024 03 22	IPTC Standards Committee	1.4	1.8

Figure 9. The specification history of the IPTC-NAA IIM and IPTC Photo Metadata Standard. There are two essential things to note: (1) The IIM uses integer Version numbers to reflect significant changes to the specification, while Revision numbers denote minor changes. The Version of a Photo Metadata Standard uses the year of initial release. From 2009 until 2016, this four-digit number features an appendix representing the month of the initial release (e.g., July 2014). In 2017, this appendix was replaced by a sequential integer starting at 1. In contrast to the IIM, Photo Metadata Standard Revision numbers do not indicate changes to the specification but merely the correction of one or more errors. From 2022, all initial releases of the Photo Metadata Standard feature a Revision number 0 (zero), indicating that nothing has been revised at that point in time. (2) The issue date of the first IIM standard is no longer well-known. Several sources provide contradicting dates, a discrepancy likely attributable to parallel work by two organisations on this standard. Personal communication with Michael W. Steidl revealed that the IPTC must have adopted the IIM standard in 1990, while ANPA took that decision later, likely in 1991.

draws upon identifiers from different namespaces, all metadata fields from the IPTC Core namespace are members of the IPTC Core schema.

The same applies to the IPTC Extension metadata schema, whose fields have identifiers from five intentionally adopted namespaces: IPTC Extension, PLUS, XMP Basic, XMP Rights Management, and Exif (respectively indicated by the prefixes *Iptc4xmpExt*, *plus*, *xmp*, *xmpRights* and *exif* in Figure 7). However, note that the *exif* prefix does not appear in Figure 7 even though the IPTC Extension metadata schema uses four GNSS-related identifiers originating from the Exif XMP namespace. These four identifiers are—amongst several other ones—part of the *Location* structure. The IPTC Photo Metadata Standard often uses such metadata field structures as the data type for some of the 62 top properties. For example, this *Location* structure consists of twelve sub-properties, four of which come from the Exif namespace. Using one or more of the *Location* structure’s sub-properties, one can provide detailed information about two top

properties: *Location created* (*Iptc4xmpExt:LocationCreated*) and *Location Shown in the Image* (*Iptc4xmpExt:LocationShown*) (see Figure 7). Similarly, the *Image Region* top property (*Iptc4xmpExt:ImageRegion*) has the *Image Region* structure as its data type. Because this IPTC *Image Region* structure lies at the centre of this paper, it will be detailed in the next section.

3. Image Regions

3.1. In With the New: IPTC Image Regions

The IPTC Photo Metadata Standard 2019.1 (IPTC Photo Metadata Working Group, 2020), released in December 2019, introduced the *Image Region* (*Iptc4xmpExt:ImageRegion*), a new top property to define one or more areas within an image and store them as disks (i.e., the region of the plane bounded by a circle) or any arbitrary *simple polygon*. Polygons are geometrical shapes bounded by a closed polyline: a curve consisting of connected line segments without any gap. Even though these line segments (also known as *edges*) may intersect, “simple polygon” denotes non-intersecting line segments (Preparata & Shamos, 1985; Schneider & Eberly,

2003). These line segments or edges meet at corners or *vertices* (singular: *vertex*), whose spatial position is described by *x* and *y* coordinates. Polygons are thus always spatially two-dimensional or 2D (Berger, 2010). As a 2D object, “polygon” refers to the polyline perimeter and the region it bounds (Preparata & Shamos, 1985; Schneider & Eberly, 2003). Some types of polygons are well known, like triangles (shapes formed by three line segments and three vertices) and rectangles, but polygons can have an arbitrary number of edges *n*. These *n*-edged polygons are called *n*-gons (Preparata & Shamos, 1985). Since a circle does not feature line segments, a disk is not a polygon.

IPTC image regions thus allow for saving disks, triangles, rectangles, or any non-intersecting *n*-gon within the image metadata. A region is stored via the *Image Region* structure (<https://www.iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-2019.1.html#image-region-structure>). Figure 10 specifies that this structure consists of:

- A *Region Boundary* structure (***Iptc4xmpExt:RegionBoundary***), which holds various fields to define the region’s measurement unit (in relative image size values or pixel count), position, and shape (rectangle, circle, or polygon). Even though a rectangle is a polygon, notice that the IPTC considers it a separate entity. Consequently, it should not be defined by the *x* and *y* coordinates of its vertices, as for a polygon, but by the *X*- and *Y*-axis coordinates of the rectangle’s upper left corner and its width and height relative to that corner. Although this representation implies that the rectangle shape cannot feature any rotation, one could always use a polygon to encode a rotated rectangle. In addition, the IPTC mixes terminology used for different spatial entities: boundaries and regions. “Circles” and “closed polylines” only refer to boundaries (i.e., excluding the enclosed region), while “disks” and “polygons” include the interior. Given that the structure is a *Region Boundary* structure, the former should be the correct terms (and then rectangle cannot be used since it refers to a specific type of four-sided polygon). Likely, the IPTC went for less mathematical rigorousness and chose terminology common in everyday non-technical

conversations. The remainder of this paper will also use “circle” and “polygon” to conform to the IPTC Photo Metadata Standard. However, it must be clear that an image region, as defined by the IPTC, comprises the pixels of the boundary and those enclosed by it.

- Four properties to store general characteristics of the image region:
 - o A *Region Identifier* (***Iptc4xmpExt:rld***), a string to uniquely identify a specific region amongst potential others within the image (the “Occurrence” column in Figure 7 shows that one image can feature multiple regions);
 - o A *Region Name* (***Iptc4xmpExt:Name***) or free-text name of the region;
 - o A *Region Content Type* (***Iptc4xmpExt:rCtype***) to define what is shown inside the region;
 - o A *Region Role* (***Iptc4xmpExt:rRole***) to indicate the region’s role among other regions of this image or other images.
- Finally, a region can have any valid XMP metadata property attached, such as a *Description* (***dc:description***) or *Keywords* (***dc:subject***). However, the property must apply to the image region, not the entire document. This feature makes image regions compelling entities, as they facilitate linking specific metadata (such as existing or future IPTC Photo Metadata properties) to designated pixel groupings.

The *Region Content Type* and *Region Role* get their value from a *controlled vocabulary*. Controlled vocabularies enforce the idea that only a limited set of terms, names, or phrases, collectively called *concepts*, can be used to describe something, thus ensuring consistency and reducing ambiguity across descriptions (Harpring, 2013; Schlegel et al., 2023). Suppose a list with terms, names, or phrases claims that “one can only use these concepts”. In that case, it is a controlled vocabulary. The IPTC Photo Metadata Standard uses so-called *Entity* or *Concept* structures to store information about the concept used (see Figure 10). The structure consists of the concept’s *Name* (e.g., “plant”) and a globally unique *Identifier* like a URI where one can retrieve the definition of that concept (e.g., <https://cv.iptc.org/newscodes/imageregiontype/plant>). To aid users, the IPTC

Property	Sub-property (L1)	Sub-property (L2)	Sub-property (L3)	Data type	Occurrence	XMP identifier	User notes
Image Region	Image Region structure				multi	lptc4xmpExt:ImageRegion	
	Region Boundary	Region Boundary structure			single	lptc4xmpExt:RegionBoundary	
		Boundary Shape		String	single	lptc4xmpExt:rbShape	Rectangle, circle or polygon
		Boundary Measuring Unit		Text	single	lptc4xmpExt:rbUnit	Pixel or relative
		X-Axis Coordinate		Decimal	single	lptc4xmpExt:rbX	
		Y-Axis Coordinate		Decimal	single	lptc4xmpExt:rbY	
		Rectangle Width		Decimal	single	lptc4xmpExt:rbW	
		Rectangle Height		Decimal	single	lptc4xmpExt:rbH	
		Circle Radius		Decimal	single	lptc4xmpExt:rbRx	
		Polygon Vertices		Region Boundary Point structure		lptc4xmpExt:rbVertices	
			X-Axis Coordinate	Decimal	single	lptc4xmpExt:rbX	
			Y-Axis Coordinate	Decimal	single	lptc4xmpExt:rbY	
	Region Identifier			String	single	lptc4xmpExt:rId	
	Region Name			Text	single	lptc4xmpExt:Name	
	Region Content Type	Entity or Concept structure			multi	lptc4xmpExt:rCtype	
		Identifier		URI	multi	xmp:Identifier	
	Region Role	Name		Text	single	lptc4xmpExt:Name	
		Identifier		URI	multi	lptc4xmpExt:rRole	
	Other Metadata Property	Name		Text	single	lptc4xmpExt:Name	
				Not defined	single	any:any [<>]	Any valid XMP property

Figure 10. All information related to the *Image Region* top property is combined into an *Image Region* structure containing five properties besides any valid XMP property (all listed in the column “Sub-property (L1)”, where L1 stands for Level 1). The “Data type” information does not follow the convention of Figure 7 but mimics the more verbose style used at <https://iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-2023.2.html>.

has created the *IPTC Region Content Type NewsCodes* (<https://cv.iptc.org/newscodes/imageregiontype>) and *IPTC Image Region Role NewsCodes* (<https://cv.iptc.org/newscodes/imageregionrole>) controlled vocabularies.

3.2. Out With the Old: MWG and Microsoft Image Regions

The possibility of defining and storing image regions was not standardised by the IPTC only. Almost a decade earlier, in 2010, the Metadata Working Group (MWG) proposed a way to deal with image regions. The MWG organisation was based on a 2006 proposal by Microsoft Corporation. In 2007, four other leading companies in the digital media industry joined, creating a consortium of five founding members: Adobe Systems Incorporated, Apple Incorporated, Canon Incorporated, Microsoft Corporation and Nokia Corporation (Metadata Working Group, 2008a).

Sony Corporation joined this variety of vendors in 2008. In September of that year, the MWG launched their first version of “Guidelines for handling image metadata”. The document specified how to prioritise Exif, IPTC-NAA IIM, and XMP-based metadata to avoid metadata conflicts and confusion due to content overlap between these commonly used standards (Metadata Working Group, 2008b). In other words, the guidelines addressed how the metadata of

digital still images should be stored and exchanged so that photographers (both amateur and professional), camera manufacturers, software developers, and service providers could achieve better compatibility and consistency in their use. A few months later, in February 2009, version 1.0.1 fixed a few grammar mistakes and reformulated or expanded some descriptions (Metadata Working Group, 2009). A substantially updated and expanded second version of these guidelines saw the light of day in November 2010, covering text encoding, hierarchical metadata, and image collections (Metadata Working Group, 2010). In addition, this 2010 MWG standard also provided guidelines on dealing with image regions.

Nowadays, nearly all photo editors and DAMs supporting image regions still stick to this standard. The MWG region information is also embedded via XMP tags, defined in the MWG Regions namespace (Metadata Working Group, 2010). This standard thus seems to have broad support. However, preferring it over the *Image Region* property of the IPTC seems sub-optimal for two reasons. First, the MWG image region recommendations only support four types of content: Face, Pet, Focus, and BarCode. If the content of the image region is not one of those four types, the *Type* field must be left empty. [Note that in 2017/2018, the IPTC tried to

contact the MWG to extend this closed *Type* list and clarify other details about these image region recommendations. Since the MWG never responded, the IPTC created their own *Image Region* property (personal communication with Michael W. Steidl)]. Second, the MWG organisation no longer exists, so their Regions namespace will not evolve anymore. Both issues make the MWG image region less

flexible and future-proof than its IPTC alternative.

The same must also be said about an attempt by Microsoft. The Microsoft Photo 1.2 namespace provides a minimal set of XMP properties to deal with image regions (Microsoft, 2021). Although this namespace was already proposed in the mid-2000s (the authors of this paper could not retrieve

Name	URI	Preferred prefix	Dereferenceable
Adobe PDF	http://ns.adobe.com/pdf/1.3/	pdf	no
Camera Raw	http://ns.adobe.com/camera-raw-settings/1.0/	crs	no
Dublin Core	http://purl.org/dc/elements/1.1/	dc	yes
Exif 2.2 or earlier	http://ns.adobe.com/exif/1.0/	exif	no
Exif 2.21 or later	http://cipa.jp/exif/1.0/	exifEX	no
IPTC Core	http://iptc.org/std/lptc4xmpCore/1.0/xmlns/	lptc4xmpCore	yes
IPTC Extension	http://iptc.org/std/lptc4xmpExt/2008-02-29/	lptc4xmpExt	yes
Microsoft Photo 1.2	https://ns.microsoft.com/photo/1.2/	MP	no
MWG Regions	http://www.metadataworkinggroup.com/schemas/regions/	mwg-rs	no
Photoshop	http://ns.adobe.com/photoshop/1.0/	photoshop	no
PLUS	http://ns.useplus.org/ldf/xmp/1.0/	plus	no
TIFF Rev. 6.0	http://ns.adobe.com/tiff/1.0/	tiff	no
XMP	http://ns.adobe.com/xap/1.0/	xmp	no
XMP Media Management	http://ns.adobe.com/xap/1.0/mm/	xmpMM	no
XMP Rights Management	http://ns.adobe.com/xap/1.0/rights/	xmpRights	no

Figure 11. An overview of the name, URI, and preferred prefix for all XMP namespaces mentioned in this paper, along with a few other common ones. Note that namespace names might differ depending on the source. For instance, the Exif and TIFF namespaces are called differently on Adobe's XMP website (<https://developer.adobe.com/xmp/docs/XMPNamespaces>) and in their official XMP document (Adobe Systems Incorporated, 2005). "Dereferenceable" means that the URI has a representation accessible through a web browser. XMP properties are commonly written in a *preferredPrefix:property* style, for example *lptc4xmpCore:Location* (see Figures 7 and 10).

the exact year of release), these image region properties are barely used outside some default Windows photo applications.

4. GRAPHIS: Under the Hood

Since its inception in September 2021, project INDIGO has sought a way to properly segment and annotate graffiti photographs. Although various software tools exist to accomplish that, they are either based on the MWG XMP properties or use proprietary ways, so results are not portable between applications. That is why INDIGO wanted

to leverage the relatively new IPTC *Image Region* property for this task. However, in 2021 and even 2022, barely any software could visualise, let alone create and save, IPTC-based image regions via a graphical interface. That is why GRAPHIS came into existence at the start of 2023.

GRAPHIS (Generate Regions and Annotations for PHotos using the IPTC Standard) is an open-source and freely available Windows-based software to create image regions, annotate them with graffiti descriptions or transcriptions, and visualise them. The backend of GRAPHIS is programmed

in Python 3 (<https://www.python.org>), while PySide—also known as Qt for Python (https://wiki.qt.io/Qt_for_Python)—was used for the Graphical User Interface (GUI). In addition, GRAPHIS relies on many other pieces of software, of which the most prominent ones function as interfaces for data handling: ExifTool (<https://exiftool.org>) to read and write photo metadata, the Python wrapper rawpy (<https://pypi.org/project/rawpy>) for LibRaw (<https://www.libraw.org>) to read the primary image pixels of RAW photo files, and the database engine SQLite (<https://www.sqlite.org>) for intermediate data storage. Luckily, thanks to GRAPHIS' GUI, one does not need to know and understand how these separate software components operate. Finally, the *GRAPHIS Image Region* vocabulary (<https://vocabs.acdh.oew.ac.at/graphis-imgreg>) provides GRAPHIS with a controlled list of concepts defined explicitly for graffiti image regions. The entire GRAPHIS source code repository is available at <https://github.com/GraffitiProjectINDIGO/GRAPHIS>, while the latest compiled release—ready for direct installation on a Windows machine—can be found at <https://github.com/>

[GraffitiProjectINDIGO/GRAPHIS/releases](https://github.com/GraffitiProjectINDIGO/GRAPHIS/releases).

Rather than detailing the individual software components of GRAPHIS, the following section will provide an overview of GRAPHIS' GUI and simultaneously explain the software's general operating principles. A comprehensive account of the *GRAPHIS Image Region* vocabulary follows in Section 4.2.

4.1. The GRAPHIS GUI: a One-Stop-Shop for Image Region Operations

4.1.1. User Info

Upon starting GRAPHIS, a welcome screen asks the user for identification in the form of a name and a non-obligatory URI, such as an ORCID (<https://orcid.org>) (see Figure 12 on the left). This information is saved, allowing one to choose an existing identification profile upon second use (see Figure 12 in the middle). GRAPHIS uses this information to populate the *Contributor* property (*lptc4xmpExt:Contributor*; <https://iptc.org/std/photometadata/specification/IPTC-PhotoMetadata#contributor>). The *Contributor* top property

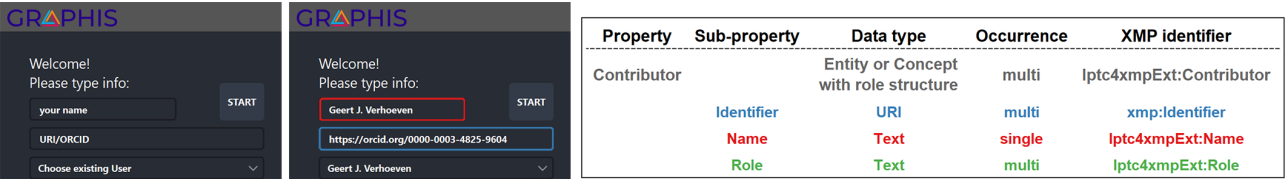


Figure 12. GRAPHIS starts by asking for user identification. One can create a new user (left) or choose an existing one (middle). This information then populates the *Name* and *Identifier* properties of the *Entity or Concept with role* structure (right), which is the structure used to define the *Contributor* property of the IPTC Photo Metadata Standard.

was introduced in November 2022 when the IPTC published the Photo Metadata Standard 2022.1 (IPTC Photo Metadata Working Group, 2023). Contributors are stored via an *Entity or Concept with role* structure (see Figures 7 and 12 on the right), a *field structure* almost identical to the *Entity or Concept* structure used in the *Region Content Type* and *Region Role* properties (see Section 3), but with an added *Role* field. Upon login, GRAPHIS uses the ORCID and user name to populate this structure's *Identifier* and *Name* fields (see Figure 12 middle and right). The *Role* field gets populated later, as its value depends on the action performed on the image region (see further).

After the initial user info screen, GRAPHIS displays its main window and directly presents two pieces of information:

- The user login in the upper right corner (see Figure 13-A);
- A welcome message in the console pane on the lower left (see Figure 13-B). Keeping an eye on the console while working with GRAPHIS pays off, as it displays error messages or auxiliary information on the finished processes.

4.1.2. Database Feedback

GRAPHIS stores all its operations on the fly in an SQLite database, a small file characterised by its .sqlite extension. This database makes it possible to quit GRAPHIS at any

point and continue later by reloading that database file. Therefore, one should first create a new database (or load an existing one) when working in GRAPHIS, operations which are all enabled via GRAPHIS' main menu. After loading the SQLite database file into GRAPHIS, new information will be displayed in three places:

- First, the database name will be shown at the top of the GUI (see Figure 13-C);
- Second, the “Database statistics” window will be updated (see Figure 13-D). For a new database, the number of images and image regions will be zero;

- Third, the console pane will announce that an *.sqlite file has been created or loaded (see Figure 13-B).

4.1.3. Adding Images

GRAPHIS supports the most common raster image file formats that store IPTC metadata: JPEG, TIFF, PNG, and many RAW formats. Images can be added to the active database anytime via the main menu. One can add all images in a folder (with or without its subfolders) or import images on a per-image basis. Each of these operations works on one file type at a time. For example, imagine a folder with *.jpg

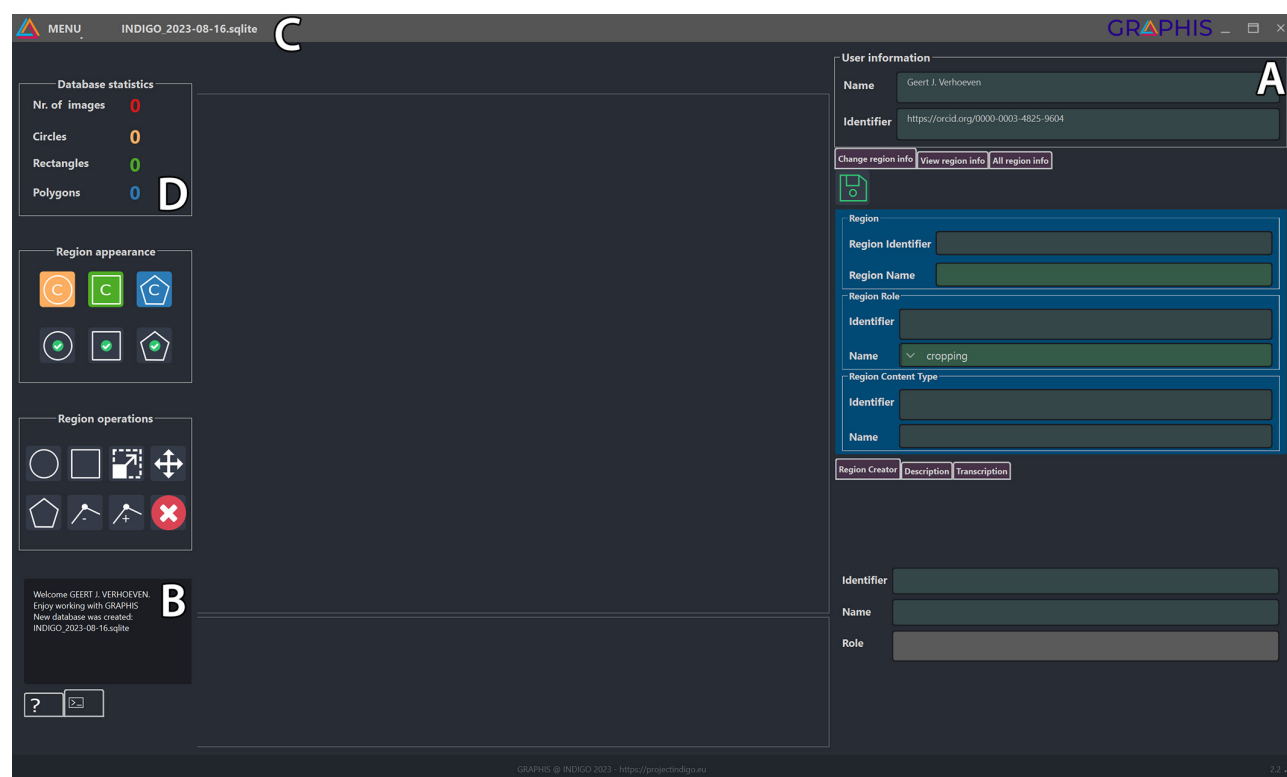


Figure 13. GRAPHIS displays general information about its status and the underlying SQLite database in four places: the user currently working with GRAPHIS (A); all the actions currently performed by GRAPHIS (B); the name of the active SQLite database (C); statistics about the image regions currently stored in the active database (D).

and *.tiff files. If the user wants to import all JPEG images into GRAPHIS, one *.jpg file must be selected. Since this operation discards all files without a .jpg extension, a second operation must be executed to import other file types, like TIFFs or RAW photos. Upon addition, the path of each

image is stored as a path relative to the folder of the *.sqlite file. Consequently, the images and database must be stored on the same drive (C:\, D:\,...), or image import will result in an error. Images already part of the database will not be added once more.

Like most DAM or MAM systems, GRAPHIS manages metadata internally by reading them from the added images and writing them into the local SQLite database (and embedding that database info into the image file during export, an operation described later). To ensure that metadata are accurately read and written, GRAPHIS relies on a stand-alone Windows executable version of ExifTool (<https://exiftool.org>). ExifTool can be considered the Swiss army knife of file metadata manipulations. It is a command-line application, but GRAPHIS' GUI means that users do not need to know how to work with ExifTool. Figure 14 shows the updated data statistics (see Figure 14-A) and console (see Figure 14-B) after loading one NEF (i.e., Nikon Electronic Format, Nikon's RAW image file format) and four JPEGs.

The images get loaded as small previews (see Figure 14-C) whose size and order cannot currently be altered; as prescribed by the IPTC, GRAPHIS discards any rotation value for the image stored in the Exif metadata. Two of the loaded JPEG images in Figure 14 are files distributed by the IPTC (a Photo Metadata reference image from <https://iptc.org/std/photometadata/examples/IPTC-PhotometadataRef-Std2023.2.jpg> and an Image Region example image from <https://www.iptc.org/std/photometadata/examples/image-region-examples>). Since both files feature image regions, GRAPHIS has read and stored that info upon import. Double-clicking a photo displays it in the main window, with its image regions on top (see Figure 14-D); above the image, GRAPHIS displays its folder path and file name (see Figure 14-E).

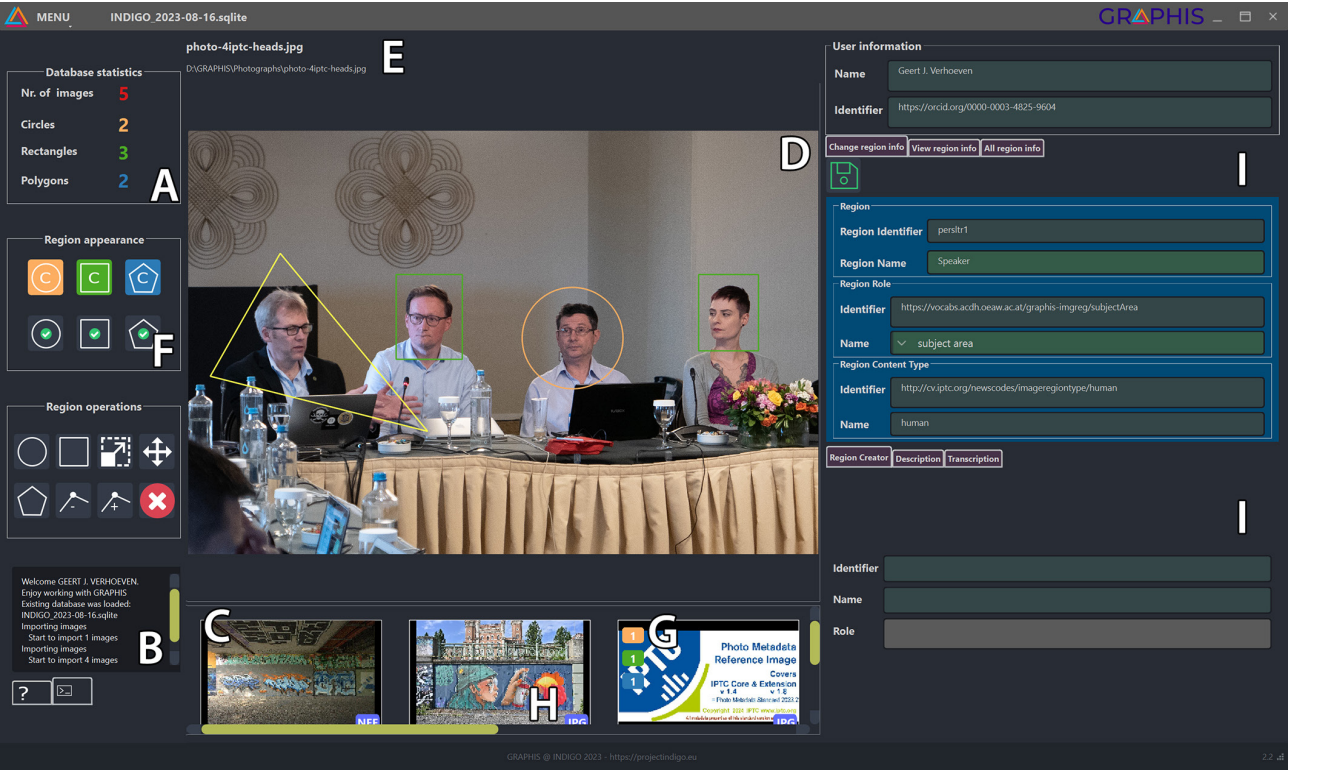


Figure 14. Importing images (with or without image regions) updates many aspects of the GRAPHIS GUI: the database statistics (A) and console (B) are refreshed; all images are rendered as small previews (C) and double-clicking one of them displays the image in large size (D) with its folder path and file name above it (E); each preview gets numbered insets (G) reflecting the number of shape-specific regions, while a small inset (H) also shows the file type of each image in the underlying database. The numbers in (A) and (G) inherit the colour settings determined in the “Region appearance” window (F). The editable properties of the selected image region are grouped below the “Change region info” tab (I).

4.1.4. Visualising Regions and Metadata

The “Region appearance” window controls the look of region boundaries via customisable colours, while the user can also hide/display one or more of those shapes (see Figure 14-F). The default colour values are defined in the `graphis.config` file, a text-based configuration file. Any change in these colours is reflected in the appearance of the numbers in the “Database statistics” window (see Figure 14-A) and in the number insets depicted on top of each preview image (see Figure 14-G). These number insets inform the user about the shape-specific region count per image. In addition, a small inset also conveys the image file type (see Figure 14-H).

Hovering over an image region turns it semi-transparent green and displays its region identifier. Double-clicking on the region turns its boundary yellow (indicating it is selected) and loads most of the image region properties on the right side of the GRAPHIS GUI in the “Change region info” tab (see Figures 14-I and 15-A). The information in these fields can be changed, as described in the following paragraph. In the “View region info” tab, all properties of the selected region can be consulted in a more structured way (see Figure 15-B). The metadata in this view are not editable, but properties can be collapsed or expanded at will. Finally, the “All region info” tab groups all the information available on each active image region, again in view-only mode but expandable or collapsible at will (see Figure 15-C).

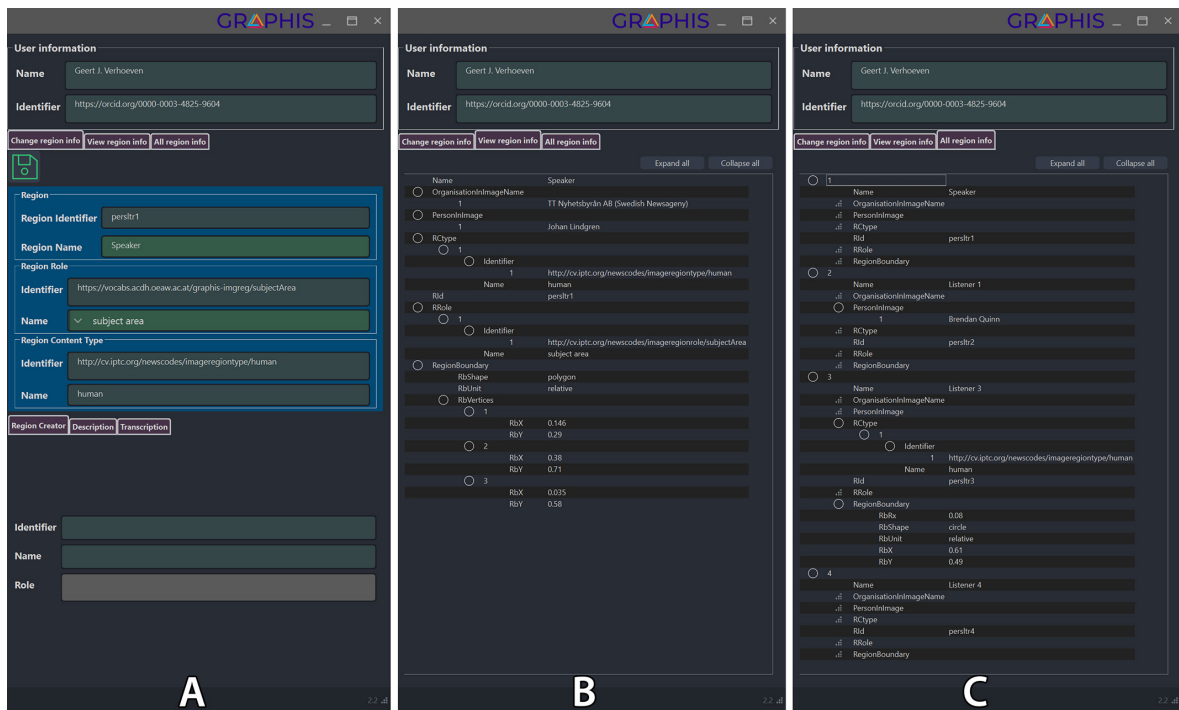


Figure 15. The editable properties of the selected image region are grouped below the “Change region info” tab (A). The “View region info” tab (B) displays all the properties of the *Image Region* structure of the selected region. The “All region info” tab (C) lists all properties of every image region in the active image. In the last two tabs, properties can be expanded or collapsed at will.

4.1.5. Creating Regions and Metadata

To create or alter an image region, the user should select one of the region operators with a left mouse click. With the circle and rectangle tools, drawing starts and ends with a right mouse click. Polygons are finished with a left click

because every right click of the mouse creates a new vertex. Upon finishing the creation of a shape, its *Region Boundary* information gets automatically stored in the SQLite database, while the *Image Region* property gets the following default values:

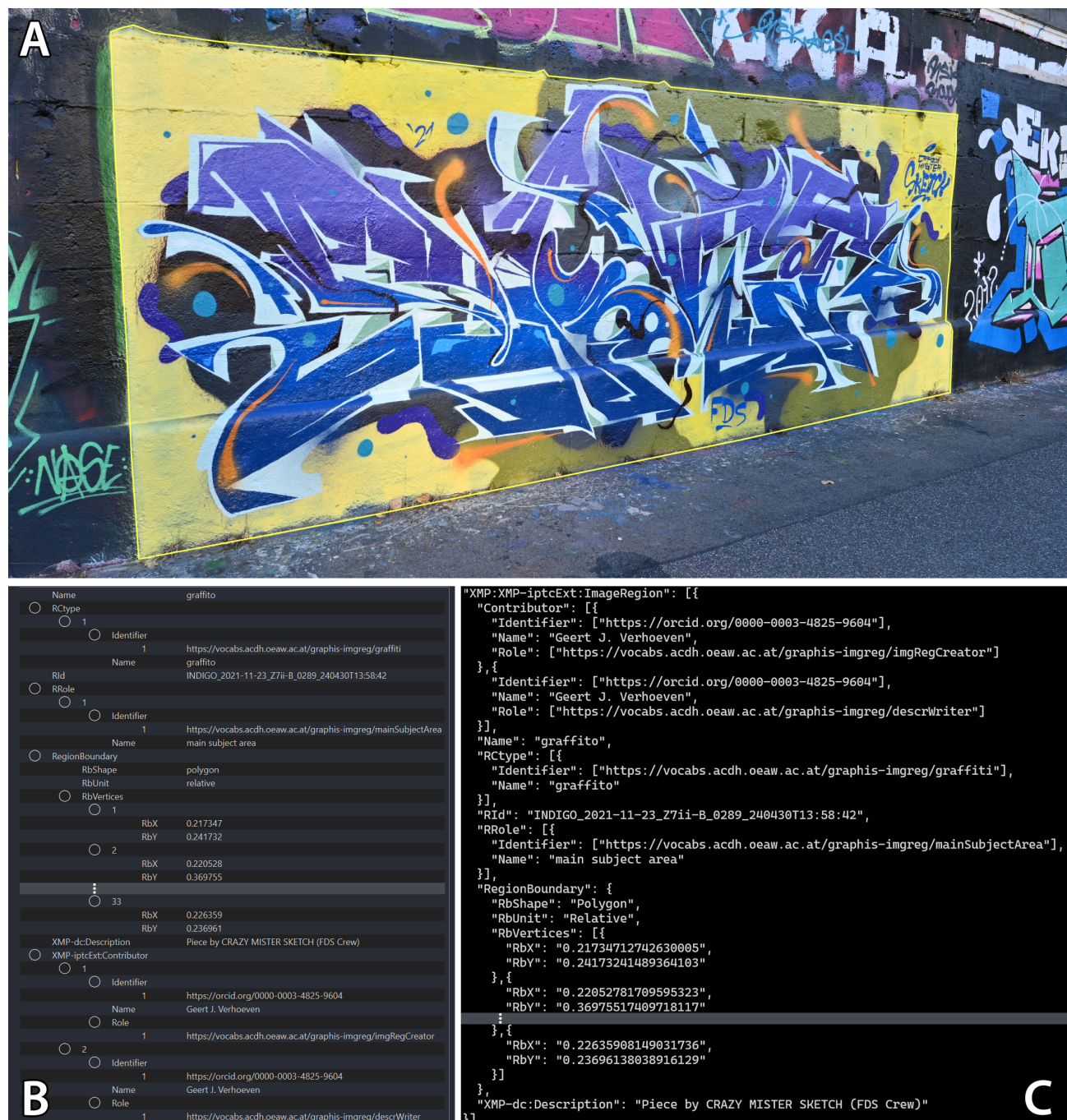


Figure 16. A 33-gon or 33-sided polygon (A) indicates the border of a graffito. The metadata of that image can be shown inside GRAPHIS (B) or visualised in the Windows command-line interface with ExifTool (C) after GRAPHIS has updated the image file metadata. For brevity, (B) and (C) only depict the coordinates of the first two and last vertices. The three dots and light grey bar indicate where the GRAPHIS and ExifTool output were partly removed.

- A *Region Identifier*, in the form of “imageFileName_YYMMDDThh:mm:ss”, where “YY” indicates a two-digit year, 00 through 99; “MM” runs from 01 through 12, denoting the month of the year; “DD” indicates a two-digit day of that month, 01 through 31; “T” indicates Time; “hh” indicates a zero-padded hour between 00 and 24; “mm” refers to a zero-padded minute between 00 and 59 and “ss” refers to a zero-padded second between 00 and 60.
- A *Region Name*, which depends upon the shape:
 - o A rectangle gets “graffito text” because a rectangle is the default shape to delineate and transcribe textual elements for most machine learning software;
 - o A polygon and a circle both get “graffito”, as a polygonal shape assumes that a graffito is outlined in detail, while a circle enables a quick indication of a graffito.
- A *Region Role Identifier* and *Region Role Name*, both coming from a controlled vocabulary. Although the IPTC has created a controlled vocabulary for this purpose (<https://cv.iptc.org/newscodes/imageregionrole>), GRAPHIS uses its own graffiti-specific controlled vocabulary (see Section 4.2). Following the reasoning of the *Region Name* property, the *Region Role Identifier* and the *Region Role Name* get default values based on the region shape:
 - o Rectangle: <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/areaOfInterest> and “area of interest”;
 - o Polygon and circle: <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/mainSubjectArea> and “main subject area”.
- A *Region Content Type Identifier* and *Region Content Type Name*. Since a controlled vocabulary must populate both fields, shape-dependent values come again from the dedicated GRAPHIS *Image Region* vocabulary (see Section 4.2):
 - o Rectangle: <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/textGraffito> and “text (graffito)”;
 - o Polygon and circle: <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/graffiti> and “graffito”.
- A *Region Creator*, stored as the *Contributor* property of the IPTC Photo Metadata Standard (*Iptc4xmpExt:Contributor*).
 - o The *Identifier* and *Name* are those used to log in;
 - o The *Role* field equals <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/imgRegCreator>, which is the URI from the GRAPHIS *Image Region* vocabulary identifying the concept “image region creator”.

Although all these values are auto-generated upon region creation, two critical comments are necessary:

- A user can set most of these predefined values in the `graphis.config` file, a text-based configuration file (but see Section 5.3 for some comments);
- Each value can be changed in the “Change region info” tab. Pressing the green floppy disk icon (see Figure 15-A) or the shortcut CTRL+S saves all changes to the database.

Descriptions or transcriptions are not auto-generated. The former is saved in the *Description* property of the IPTC Extension schema (**dc:description**) (see Figure 16-B & C), while the latter gets stored in the *Title* property of the IPTC Core schema (**dc:title**) (see Figure 17-B & C). When a description or transcription is added, the current user information populates a new *Contributor* property, but now the *Role* field gets either the “description writer” URI (i.e., <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/descrWriter>; see Figure 16-B & C) or the “transcript writer” URI (i.e., <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/transcrWriter>; see Figure 17-B & C) of the GRAPHIS *Image Region* vocabulary (see Section 4.2). Several contributors, each with a different role, can be stored per region as the IPTC has allowed multiple values for this property (see Figure 7). However, GRAPHIS only stores one entry per contributor role. For instance, if Person A creates an image region and writes its description, Person A is stored as the image region creator and the description writer. If Person B opens the database and alters that description, Person B becomes the description writer, overwriting Person A. However, Person A remains the creator of the image region until another user adapts the region's shape.

When drawing a region, the last-used drawing tool remains active (indicated by its yellow icon), allowing the user to keep drawing with the same tool without activating it

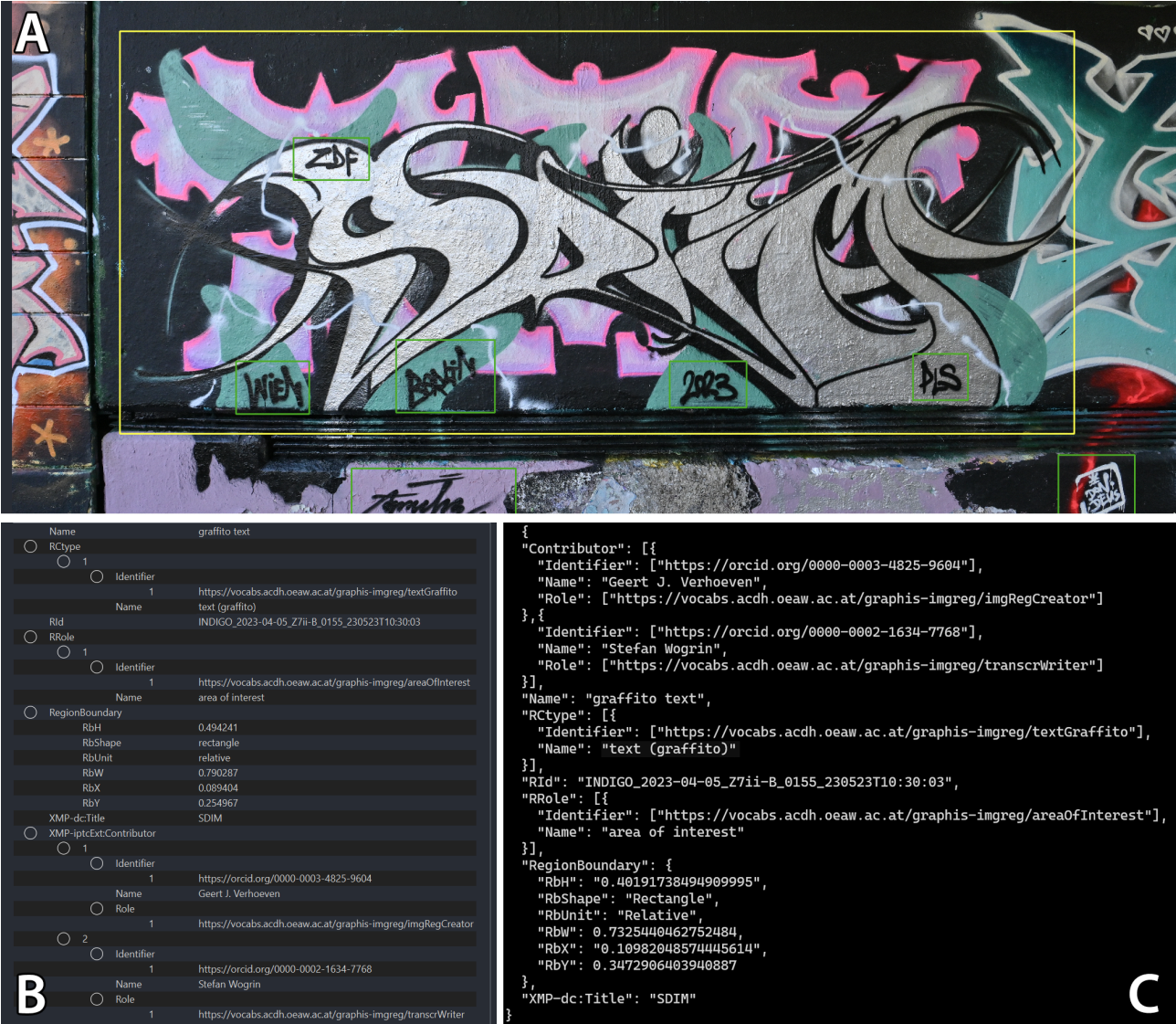


Figure 17. Various rectangles indicate regions with text on a photo of a graffitied wall (A). The metadata of the active image region can be shown inside GRAPHIS (B) or, after saving the region metadata within the image, visualised via ExifTool (C).

every time. Suppose metadata properties need to be added (like a description) or changed (like the region name) after drawing that region. In that case, the user can quickly save that new information with CTRL+S, avoiding a mouse click on the saving icon. The Enter and Backspace buttons allow navigation to the next or previous image. However, this will only work if the cursor is not on a metadata field, as one would otherwise start typing in that field. These features support the rapid creation of polygons with minimal mouse

clicks. GRAPHIS also ensures that a region cannot be finalised if it partly lies outside the image boundaries. In addition, intersecting polygon edges are not allowed.

Once an image region has been created, it is possible to modify its shape in various ways using some of the tools provided by GRAPHIS. For example, circles and rectangles can be resized, while the same tool can also be used to move polygon vertices. Other tools allow the user to shift

or delete the entire shape, or to add or remove individual polygon vertices. As with the drawing tools, a left mouse click activates these shape modification tools, while a right mouse click executes the action.

4.1.6. Saving Results

Photo metadata management in software typically involves extracting embedded metadata from an image upon file import, assigning extracted field values to corresponding metadata pane fields, changing or adding metadata values, and re-embedding these updated values back into the image file upon saving or exporting. Thanks to ExifTool, GRAPHIS can also store the (newly created or altered) image regions and their annotations back into the original images at any time. GRAPHIS either saves all metadata to the original image files, or into a backup of these files. If something goes

wrong, the latter option ensures that the original image files stay unharmed. Feedback on this process is again provided in the console window. In its current version (i.e., 2.2), GRAPHIS embeds metadata into every image file, even RAW photographs. Section 5.2 will explain why this behaviour should be changed in the future.

Finally, “Save bounding boxes to CSV file” is a function that creates a Comma-Separated Values file of all the image regions in the database. This *.csv file contains image region information useful for machine learning purposes (see Figure 18). In machine learning, image labelling/tagging/annotating software is used to draw bounding rectangles around objects and attach labels to them. The coordinates of these rectangles and their labels are then typically exported into a text-based *.csv file. Although many free

Image	Type	Rid	UpperLeftX	UpperLeftY	Width	Height	XMP-dc:Description	XMP-dc:Title	Polygon vertices
INDIGO_2023-04-05_Z7ii-B_0002.jpg	rectangle	INDIGO_2023-04-05_Z7ii-B_0002_230521T23:10:30	1047	3492	1186	1025		CRIME AFTER CRIME	[[1794.417884031533, 1355.5133833927827], [1820.6776579441894, 2060.150660049068], [1886.3270927258309, 3158.6845353952017], [1916.9634956238302, 3845.8152861097237], [1957.6727272727271, 4415.70909090909], [3159.9261274896758, 4222.20537887812], [4287.84966370481, 3990.24040426283415], [5497.046005716127, 3745.152819444536], [6713.748863669214, 3489.4251933616383], [6726.878750625543, 3110.541616555332], [6713.748863669214, 3066.773267009043], [6687.489089796556, 3049.268810791325], [6696.242347727444, 2874.2036313414226], [6713.748863669214, 2590.3852339528257], [6713.748863669214, 2414.6576078699313], [6735.632008596429, 2213.3326745395643], [6748.761895552758, 2051.397402078182], [6748.761895552758, 1924.475161500342], [6748.654545454545, 1813.7416181818181], [8204.50909090909, 1751.272727272727], [5076.8896201136205, 1679.3838383155473], [5037.499962244636, 1644.370906432005], [5006.863559346537, 1657.500793388335], [4722.382675292758, 1626.8643904902342], [4634.85009583903, 1639.9842774466021], [4339.122469501008, 1613.745453533956], [4332.86269586302, 1578.721471650364], [4280.343147763038, 1596.227967592135], [3400.640721689027, 1495.5655206269512], [3194.9391593732175, 1486.8122629560658], [2424.652457935291, 1412.4095702035388], [2207.854545454545, 1362.254545454545], [1930.0933825802585, 1316.1237325237976], [1868.8205767840595, 1329.2536194801262]]
INDIGO_2023-04-05_Z7ii-B_0017.jpg	rectangle	INDIGO_2023-04-05_Z7ii-B_0017_230523T10:10:49	5502	3624	546	389		YUZI	
INDIGO_2023-04-05_Z7ii-B_0035.jpg	polygon	INDIGO_2023-04-05_Z7ii-B_0035_230521T23:16:14	1794	1316	4954	3099	Piece and character by JAKOB		
INDIGO_2023-04-05_Z7ii-B_0035.jpg	circle	INDIGO_2023-04-05_Z7ii-B_0035_230521T23:16:29	353	659	1495	1801			[[1794.417884031533, 1355.5133833927827], [1820.6776579441894, 2060.150660049068], [1886.3270927258309, 3158.6845353952017], [1916.9634956238302, 3845.8152861097237], [1957.6727272727271, 4415.70909090909], [3159.9261274896758, 4222.20537887812], [4287.84966370481, 3990.24040426283415], [5497.046005716127, 3745.152819444536], [6713.748863669214, 3489.4251933616383], [6726.878750625543, 3110.541616555332], [6713.748863669214, 3066.773267009043], [6687.489089796556, 3049.268810791325], [6696.242347727444, 2874.2036313414226], [6713.748863669214, 2590.3852339528257], [6713.748863669214, 2414.6576078699313], [6735.632008596429, 2213.3326745395643], [6748.761895552758, 2051.397402078182], [6748.761895552758, 1924.475161500342], [6748.654545454545, 1813.7416181818181], [8204.50909090909, 1751.272727272727], [5076.8896201136205, 1679.3838383155473], [5037.499962244636, 1644.370906432005], [5006.863559346537, 1657.500793388335], [4722.382675292758, 1626.8643904902342], [4634.85009583903, 1639.9842774466021], [4339.122469501008, 1613.745453533956], [4332.86269586302, 1578.721471650364], [4280.343147763038, 1596.227967592135], [3400.640721689027, 1495.5655206269512], [3194.9391593732175, 1486.8122629560658], [2424.652457935291, 1412.4095702035388], [2207.854545454545, 1362.254545454545], [1930.0933825802585, 1316.1237325237976], [1868.8205767840595, 1329.2536194801262]]
INDIGO_2023-04-05_Z7ii-B_0035.jpg	rectangle	INDIGO_2023-04-05_Z7ii-B_0035_230521T23:15:44	1408	2265	321	181		SAES.	

Figure 18. An example of a *.csv output file (slightly formatted to improve presentation).

and payware labelling solutions exist, GRAPHIS can also be used to accomplish this, with the added benefit that the bounding rectangles and their labels can be stored in the images according to a prevailing photo metadata standard. For polygons or circles, GRAPHIS will compute and export the coordinates of the smallest rectangle that encompasses that shape, the so-called *minimum bounding rectangle*.

4.2. The GRAPHIS Image Region Vocabulary

As mentioned in Section 3.1, image regions can be annotated to provide further information about what they depict or why they were created. This can be done using the *Region Content Type* (*lptc4xmpExt:rCtype*) and *Region Role* (*lptc4xmpExt:rRole*) properties, which store the information in an *Entity* or *Concept* structure. This structure consists of a

concept's *Name* (*Iptc4xmpExt:Name*) and a URI as a unique *Identifier* (*xmp:Identifier*) for that concept (see Figure 10). For this purpose, the IPTC has published two controlled vocabularies which provide a set of predefined concepts for both properties: the *Image Region Type* vocabulary (<http://cv.iptc.org/newscodes/imageregiontype>) and the *Image Region Role* vocabulary (<http://cv.iptc.org/newscodes/imageregionrole>). For example, the *Image Region Type* vocabulary contains concepts such as “animal”, “artwork”, “human”, and “rock formation”, while “recommended cropping”, “subject area”, “area of interest”, and “business” are concepts of the *Image Region Role* vocabulary. Both vocabularies are maintained as part of the IPTC NewsCodes (<https://cv.iptc.org/newscodes>) and may be updated with new concepts over time. The use of these controlled vocabularies is not mandatory but recommended.

Both the *Region Role* and the *Region Content Type* fields are available in GRAPHIS. They can be found in the right pane when editing the metadata of an image region, under the headings “Region Role” and “Region Content Type” (see Figure 15-A). For the *Region Role* property, a user can either enter a URI by typing it into the “Identifier” field, or select one of the predefined terms from the drop-down menu next to “Name” (which will automatically fill in the corresponding value for “Identifier”). This drop-down menu contains three terms defined by the IPTC *Image Region Role* vocabulary: “area of interest”, “main subject area”, and “subject area”. As mentioned in Section 4.1.5, creating a new region assigns default values to *Region Role* fields. These default values can be set in the *graphis.config* file, but any initial value can be changed later.

Setting the values for the *Region Content Type* fields works similarly, except for the lack of a drop-down menu (which is something the next version of GRAPHIS should solve). In addition, the automatically assigned default values do not come from the recommended IPTC *Image Region Type* vocabulary, since the concepts of that vocabulary do not cover various use cases encountered in project INDIGO. For example, there are many reasons why it is helpful to define which part of a photo contains a specific graffiti, or where textual graffiti elements are present (see Section 5.1).

The controlled vocabulary used by GRAPHIS should thus include concepts to indicate these types of image regions. In addition, INDIGO wanted to go one step further and exploit the full potential of the *Contributor* property of the IPTC Photo Metadata Standard (*Iptc4xmpExt:Contributor*) (see Sections 4.1.1 and 4.1.5). This property helps define the role in which a particular actor (such as a person or organisation) has contributed to an image. The IPTC created a controlled vocabulary for this field too, the *Content Production Party Role* vocabulary, which can be accessed at <http://cv.iptc.org/newscodes/contentprodpartyrole> (as with the previous examples, use of this vocabulary is not mandatory). The concepts in this vocabulary include “Author”, “Description Writer”, and “Generative AI Prompt Writer”, to name a few. However, no concept defines the “creator of an image region”. The inability to assign such a value to a specific actor makes it difficult to trace the provenance of image regions. Also, when textual elements are present in an image, someone might write a transcription of that text and include it in the image metadata. To record who created this transcription, a controlled vocabulary should also include a value such as “transcript writer”.

Therefore, project INDIGO decided to create a new controlled vocabulary comprising both selected values from the IPTC vocabularies (i.e., only those values strictly necessary for the purposes of GRAPHIS) and the additional values created within the scope of INDIGO. The resulting product, the *GRAPHIS Image Region* vocabulary, is hosted on the Vocab service of ACDH-CH (Austrian Centre for Digital Humanities and Cultural Heritage, one of the institutions involved in project INDIGO) and can be accessed with a web browser via its URI <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg>. This URI automatically redirects to the presentation page https://vocabs.acdh.oeaw.ac.at/graphis_imgreg_browse/en (see Figure 19).

The vocabulary was formalised according to a widely used RDF-based data model, the Simple Knowledge Organization System or SKOS (Miles & Bechhofer, 2009). SKOS enables the definition of controlled vocabularies, even when they include hierarchical and associative relationships between the concepts, thus allowing the construction of very complex

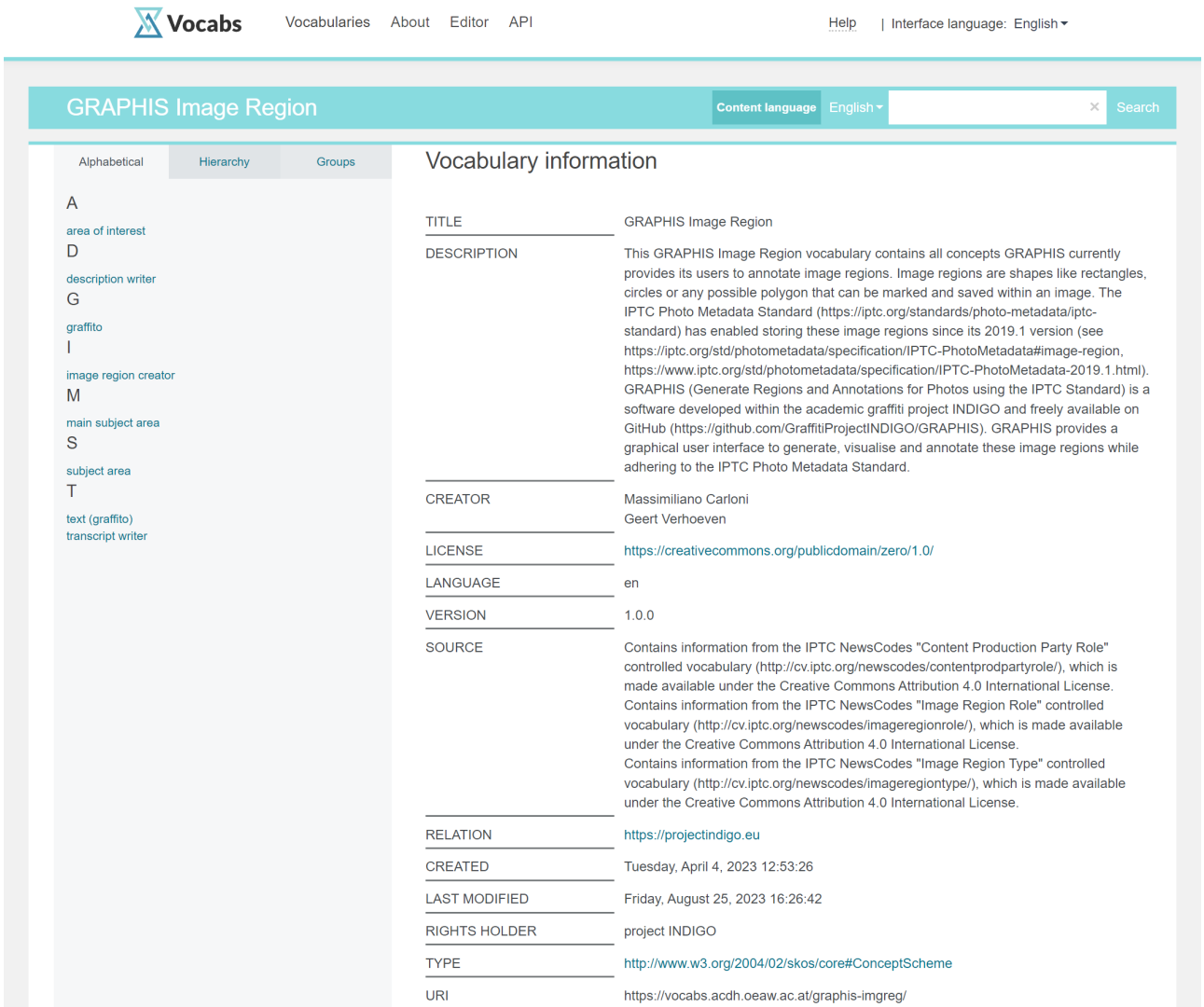


Figure 19. The main page presenting the *GRAPHIS Image Region* vocabulary (https://vocabs.acdh.oeaw.ac.at/graphis_imgreg_browse/en) shows the vocabulary title as well as additional metadata fields such as “description”, “creator”, and “license”. Individual concepts can be accessed by clicking on them in the left pane. The default view of the concept list is “Alphabetical”, but other options are available (see also Figure 20).

vocabularies such as thesauri (see Schlegel et al. (2023) for more details). SKOS also facilitates the inclusion of terms in different languages that refer to the same concept, as well as the establishment of mapping relationships that create ‘matches’ between concepts of different vocabularies. For example, if the concept “graffiti” defined in one vocabulary can be considered equivalent to the concept “graffiti” in

another vocabulary (such as the Getty Art & Architecture Thesaurus; <http://vocab.getty.edu/aat>), a relationship of the kind *skos:exactMatch* can be established between the first and the second concept.

In the case of the *GRAPHIS Image Region* vocabulary, the concepts are organised according to a flat hierarchy: all

concepts are listed one after the other, and all are on the same hierarchical level because no concept is ‘narrower’ (i.e., a sub-concept) than another (see Figure 20 on the left, where the “Hierarchy” tab is selected). However, to provide a clear view for users, values related to a specific metadata field have been grouped using another SKOS-

specific construct, i.e. *skos:Collection*. The grouping of the concepts by collections can be viewed by clicking on the “Groups” tab (see Figure 20 on the right). Each of the three collections corresponds to the metadata property for which the contained values (visible when expanding the collection) can be used.

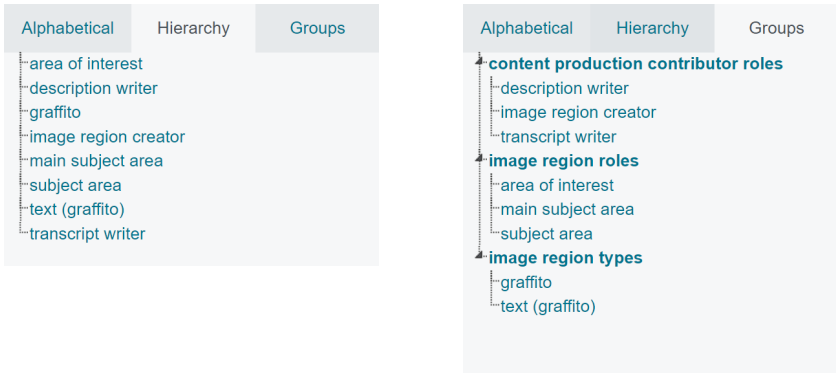


Figure 20. Vocab can display the concepts of a controlled vocabulary according to different criteria. The “Hierarchy” tab (on the left) shows the hierarchical relationships between the concepts; however, in the case of the *GRAPHIS Image Region* vocabulary, all concepts are placed on the same hierarchical level. The “Groups” tab (on the right) shows the same concepts according to their collection (collection names are in bold; all collections have been expanded by clicking on the small arrow next to them).

As shown in Figure 20 (on the right), the *content production contributor roles* collection contains three concepts relevant to the *Contributor* property: “description writer” comes from the *IPTC Content Production Party Role* vocabulary, whereas “image region creator” and “transcript writer” have been created specifically for GRAPHIS. Within the *image region roles* collection, there are three concepts (“area of interest”, “main subject area”, and “subject area”), all taken from the *IPTC Image Region Role* vocabulary. Finally, the collection of *image region types* is populated by two concepts related explicitly to graffiti research: “graffiti” and “text (graffiti)”.

Metadata are available for the entire vocabulary as well as for individual concepts. The main page shown in Figure 19 includes general vocabulary information, while clicking on a concept in the left pane displays the metadata associated

with that concept (e.g., “text (graffiti)”, shown in Figure 21). These concept-specific metadata include: a preferred term (i.e., the preferred way to refer to this concept) in English; a definition (which, for concepts borrowed from the IPTC, was derived from the respective IPTC vocabulary with appropriate attribution); additional documentary notes (if available); and the URI of the concept. Metadata for collections are also included in the vocabulary and can be viewed by clicking on a collection after selecting the “Groups” tab in the left pane.

These concept URIs and their respective preferred terms are used in GRAPHIS; for example, when selecting the term “area of interest” for the “Name” field of the *Region Role* property, the “Identifier” field is automatically filled with the URI <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/areaOfInterest>. The authors of this paper (who also authored

PREFERRED TERM	text (graffito) 
DEFINITION	Any alphanumerical character as part of a graffito, easy to read or not.
BELONGS TO GROUP	image region types
URI	https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/textGraffito 
DOWNLOAD THIS CONCEPT:	RDF/XML Turtle JSON-LD Created 4/4/23, last modified 5/5/23

Figure 21. This is an example of a presentation page for a single concept, in this case “text (graffito)”, accessible at https://vocabs.acdh.oeaw.ac.at/graphis_imgreg_browse/en/page/textGraffito. Entering the URI of the concept (<https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/textGraffito>) in a web browser automatically redirects to this page.

the GRAPHIS vocabulary) decided to create new URIs in the <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/> namespace for concepts—like “area of interest”—already existing in the IPTC vocabularies. This is why GRAPHIS uses <https://vocabs.acdh.oeaw.ac.at/graphis-imgreg/areaOfInterest> instead of <http://cv.iptc.org/newscodes/imageregionrole/areaOfInterest> as the URI for “area of interest”. However, if applicable, each concept still includes the original IPTC URI in the “exactly matching concepts” metadata field (based on the *skos:exactMatch* property). In addition, the preferred terms for the GRAPHIS concepts closely mirror their corresponding IPTC concepts (when present), except that all are written in lowercase for consistency.

Recreating URIs and metadata for the IPTC concepts directly in the *GRAPHIS Image Region* vocabulary has a few advantages. Since one source contains all information relevant to GRAPHIS (i.e., the vocabulary as modelled in Vocab), adding, altering, or accessing concepts needed for GRAPHIS is easier. At the same time, those concepts also feature consistent URIs. Finally, this solution avoids visualisation issues in the Vocab service and potential semantic conflicts due to future updates of the IPTC vocabularies. Applications other than GRAPHIS can reuse these concepts by indicating the URIs assigned to them in the *GRAPHIS Image Region* vocabulary. Metadata about the concepts (including mapping relationships to external resources) can be retrieved directly from the concept’s URI if the requesting application supports RDF, or via the API of Skosmos (<https://skosmos.org>), the open-source software

on which the Vocab service is based. To facilitate reuse, the *GRAPHIS Image Region* vocabulary has been released under a CC0 public domain licence (<https://creativecommons.org/publicdomain/zero/1.0>).

5. GRAPHIS: Considerations

5.1. INDIGO Workflow

GRAPHIS is generally helpful in any workflow where users want to attach specific metadata to one or more regions of an image. This is especially true for graffiti photos as they usually depict multiple graffiti. GRAPHIS allows graffiti scholars to annotate each graffito in an image with a general description and a unique identifier. Within project INDIGO, GRAPHIS also became an essential part of the processing pipeline developed to compute each graffito’s surface area and track its existence in time. To that end, the spatial extent of each graffito (as depicted by an overview photograph) can be defined with a polygon within GRAPHIS. However, this polygon has 2D pixel coordinates defined relative to the image. As a result, one cannot use this polygon to compute the surface area of the physical graffito. This is where AUTOGRAF—INDIGO’s bespoke software for orthorectifying graffiti photographs (Wild et al., 2022; Wild et al., 2023)—enters the workflow.

AUTOGRAF reads the polygon vertex coordinates saved by GRAPHIS in the photo and projects those vertices onto a georeferenced triangle-based mesh that digitally represents the graffito surface in 3D. Using photogrammetric and computer vision principles, AUTOGRAF can extract this

digital 3D surface mesh for every graffito from the series of overlapping photographs acquired during INDIGO's photo tours (Verhoeven et al., 2023). Since every point on this meshed, digital 3D surface features accurate 3D coordinates expressed in a standard coordinate reference system for East Austria (MGI/Austria GK East, EPSG:31256; <https://epsg.io/31256>), it is possible to end up with exact real-world 3D coordinates (x, y, and z) for each projected polygon vertex. AUTOGRAF thus turns the 2D image polygon defined within GRAPHIS into a 3D shape bounded by a polyline with real-world 3D coordinates (see Figure 22). Since the area of this 3D digital surface approaches the real-world area occupied by the graffito, it can be computed and stored as metadata for the real graffito (see Figure 4).

In the future, the authors also hope to use these 3D polylines to automatically find overlaps between graffiti and—depending on the photography date—temporally stamp how long each graffito segment was visible (with a lower and upper bound). The contribution by Verhoeven, Schlegel, & Wild in this volume provides more details on this idea.

Finally, GRAPHIS has also been used to create rectangles around verbal graffiti and annotate them with transcriptions (see Figure 17). It is straightforward to export those results into a *.csv file (see Figure 18), which can be input for machine learning systems trained to read graffiti automatically.

5.2. JPEG or TIFF Versus RAW+XMP

Anno 2024, XMP is the industry standard for storing metadata in the image or as separate *.xmp sidecar files. Where those metadata are stored is file-dependent. XMP data are embedded within the image file for JPEG, PNG, TIFF, PSD, PSB, DNG, GIF, PDF and a few other file formats. Formats without support for embedded XMP must store the XMP metadata in a separate but associated *.xmp sidecar file. The same holds for RAW photographs. Although these files can have an embedded XMP metadata record, an unwritten rule in the photo industry stipulates leaving RAW files untouched, hence the need for a sidecar *.xmp file with the same name as the original RAW photo.



Figure 22. AUTOGRAF can compute a 3D surface mesh (A) with texture (B) from the series of overlapping photos acquired per graffito. On this textured mesh, the GRAPHIS image polygon can be projected (C) to yield a 3D closed polyline whose vertices have real-world 3D coordinates.

The only way to avoid metadata sidecar files is not to use RAW file formats. However, this is unattainable from an academic point of view as RAW is the only scientifically justifiable photo file format to store the initially acquired

photographs (Verhoeven, 2010). However, the RAW format is not all roses. Even though most dedicated digital cameras support saving RAW files, these files feature proprietary structures and manufacturer-specific extensions. Adobe also

launched its open-source Digital NeGative (DNG) format in 2004 (Adobe Systems Incorporated, 2004), attempting to standardise the RAW file format. However, most camera manufacturers refrain from implementing it. On the positive side, *.dng files are the exception to the “do not embed XMP metadata in RAW files” mantra because their file structure was designed to carry embedded metadata.

GRAPHIS version 2.2 (i.e., the most recent version at the time of writing) does not support the creation of sidcar *.xmp files. Even though LibRaw and ExifTool enable GRAPHIS to visualise and process RAW files, the authors, for now, advise users to avoid working with RAW files unless they are in the DNG file format.

5.3. Issues and Improvements

GRAPHIS and the IPTC Photo Metadata Standard on which it is based currently only allow the creation of *simple polygons*, i.e. planar regions enclosed by a single closed polyline that does not intersect itself. Simple polygons do not allow holes, but this lack of holes can be problematic when, for instance, one needs to denote all the pixels of the character “O” sprayed without a background. Figure 23 depicts three graffiti for which a *polygon with holes* (O'Rourke, 1987) would be necessary to indicate the image pixels that belong exclusively to each graffito. Since the IPTC wants to keep the Photo Metadata Standard and its implementation in software as straightforward as possible, there are no plans to support polygons with holes (personal communication with Michael W. Steidl). Users who need image regions with



Figure 23. Three examples of a graffiti featuring spray-painted characters without a dedicated background. The spaces between those characters are thus not a part of the graffito and should, *stricto sensu*, be excluded from the image region polygon.

holes must thus develop creative solutions. One possibility could be to define the maximum extent of the entire graffito with one polygon, and delineate each hole with a separate polygon. Afterwards, those hole-indicating polygons could be subtracted from the overall polygon in dedicated software. However, it must be clear that the IPTC *Image Region* property cannot correctly store the resulting polygon with holes.

Since GRAPHIS is not meant for image processing, there is no risk of violating the integrity of the image region metadata. However, manufacturers of image processing software must take measures to avoid invalidating the coordinate definitions of the image regions when resizing, rotating, straightening, and cropping the images (or any other operation that changes the mapping of the input

pixels to the output pixels, such as the correction of lens distortion or perspective). Suppose those software packages do not adequately update the image region metadata in an automated manner. In that case, all the region boundary coordinates become nonsensical and will no longer represent the initially defined image region(s).

Users must also be aware that GRAPHIS is not meant to be a DAM or MAM replacement, meaning it lacks functions to monitor file locations or check metadata integrity. Altering image regions outside of GRAPHIS while the SQLite database still holds image region metadata that are not written back into the image file, or changing the relative location of the SQLite database versus the image files, will lead to errors. Besides the correction of some minor issues and the necessary support for sidcar *.xmp files (see Section 5.2),

there are four prominent features that GRAPHIS still would need to make the software even more straightforward to use and beneficial for various image-centric fields besides the graffiti community:

- The GUI should support removing images from the SQLite database and facilitate sorting and resizing the previews;
- The GUI should allow the annotation of image regions with any valid property of the IPTC Core and Extension schemas, not just the *Iptc4xmpExt:Contributor*, *dc:description*, and *dc:title* properties. For example, the image region metadata of the IPTC test image in Figure 15 (B & C) show how the *Iptc4xmpExt:PersonInImage* and *Iptc4xmpExt:OrganisationInImageName* fields can function as additional metadata properties;
- The GUI should automatically fetch the preferred terms and corresponding URIs from the *GRAPHIS Image Region* vocabulary. Any change in the latter would then automatically be reflected in the GRAPHIS GUI. Users could also sync GRAPHIS with the latest terms and URIs defined by another controlled vocabulary simply by specifying the URI of that vocabulary. This leads directly to the next improvement;
- The *graphis.config* file should be slightly restructured for ease of use and expanded with the base URI of a controlled vocabulary plus a custom pattern for the *Region Identifier*.

With the GRAPHIS source code freely available on GitHub (<https://github.com/GraffitiProjectINDIGO/GRAPHIS>), the authors hope that other projects or individual developers will help to implement such features.

6. Conclusion

Image file metadata are typically applied to the entire image content. Nevertheless, applying metadata to specific parts of the image can be essential for many images, such as photographs of graffiti. Defining image regions and annotating them with metadata should, however, follow specific requirements to make these regions useful, transferrable between software packages, and suitable for inventorying purposes. This paper proposed GRAPHIS, a tool to visualise, create and annotate image regions based on

the IPTC Photo Metadata Standard. As with every software package developed within project INDIGO, GRAPHIS is freely available. In this way, the authors hope GRAPHIS gets adopted within the academic (graffiti) community and integrated into existing image annotation workflows. In addition, GRAPHIS' open-source nature makes it more likely that a few enthusiastic developers will implement improvements to steadily increase its user-friendliness and relevance for various non-graffiti-specific use cases.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgements

INDIGO was funded by the Heritage Science Austria programme of the Austrian Academy of Sciences (ÖAW). The authors are very grateful to Michael W. Steidl (former IPTC Managing Director and current co-lead of the IPTC Photo Metadata Working Group) and David Riecks (current co-lead of the IPTC Photo Metadata Working Group) for their insights on the history of the IPTC Photo Metadata Standard and the discussions about the use of image regions.

References

- Adobe Systems Incorporated. (1991-2008). *Photoshop Image Resources*. https://web.archive.org/web/20090306165311/https://www.adobe.com/devnet/photoshop/psir/ps_image_resources.pdf
- Adobe Systems Incorporated. (2001a). *Adobe® XMP. Adobe Extensible Metadata Platform integration technology built on W3C standards*.
- Adobe Systems Incorporated. (2001b, September 14). *XMP – Extensible Metadata Platform. Version 1.5*. <https://xml.coverpages.org/XMP-MetadataFramework.pdf>
- Adobe Systems Incorporated. (2002). *Adobe Solutions Network: Developer Knowledgebase. Known Issue: The metadata framework name was changed from XAP to XMP*.

<http://web.archive.org/web/20020307124612/http://support.adobe.com/devsup/devsup.nsf/docs/51840.htm>

Adobe Systems Incorporated. (2004). *Digital Negative (DNG) Specification. Version 1.0.0.0. September 2004.*

Adobe Systems Incorporated. https://web.archive.org/web/20041204034106/http://www.adobe.com/products/dng/pdfs/dng_spec.pdf

Adobe Systems Incorporated. (2005). *XMP specification. September 2005.*

Adobe Systems Incorporated. (April 2012). *XMP specification part 1. Data model, serialization, and core properties.* <https://github.com/adobe/XMP-Toolkit-SDK/blob/main/docs/XMPSpecificationPart1.pdf>.

Adobe Systems Incorporated, & Knoll, T. (1990-2019). *Adobe Photoshop File Formats Specification.* <https://www.adobe.com/devnet-apps/photoshop/fileformatashtml>.

Baca, M., Harpring, P., Lanzi, E., McRae, L., & Whiteside, A. (2006). *Cataloging cultural objects: A guide to describing cultural works and their images.* American Library Association.

Berger, M. (2010). *Geometry Revealed: A Jacob's Ladder to Modern Higher Geometry.* Springer. <https://doi.org/10.1007/978-3-540-70997-8>.

Borgman, C. L. (2015). *Big Data, Little Data, No Data: Scholarship in the Networked World.* The MIT Press.

Bryson, B. (2008). *Bryson's dictionary for writers and editors.* Broadway Book.

Camera & Imaging Products Association (2023, May 29). *CIPA DC-008-Translation-2023. Exchangeable image file format for digital still cameras: Exif Version 3.0 (CIPA DC-008-2023 / JEITA CP-3451F).* CIPA-JEITA. https://www.cipa.jp/std/documents/download_e.html?DC-008-Translation-2023-E.

Chapman, C., & Brailsford, D. F. (2001). Navigating a corpus of journal papers using Handles. In A. C. Hübler, P. Linde, & J. W. T. Smith (Eds.), *Electronic publishing '01: 2001 in the digital publishing odyssey. Proceedings of the 5th ICC/IFIP conference held at the University of Kent at Canterbury, Kent, United Kingdom, 5-7 July, 2001* (pp. 249–255). IOS Press. https://elpub.architexturez.net/system/files/pdf/200123.content_0.pdf.

Comité International des Télécommunications de Presse, & Newspaper Association of America. (July 1999). *IPTC-NAA Information Interchange Model Version 4. Revision 1.* <https://iptc.org/std/IIM/4.1/specification/IIMV4.1.pdf>.

Gartner, R. (2016). *Metadata: Shaping Knowledge from Antiquity to the Semantic Web.* Springer International Publishing. <https://doi.org/10.1007/978-3-319-40893-4>.

Harpring, P. (2013). *Introduction to controlled vocabularies: Terminology for art, architecture, and other cultural works* (Updated edition). Getty Research Institute.

Horodyski, J. (2022). *Metadata matters.* CRC Press.

International Organization for Standardization (2016, April 7). *Information and documentation – Records management – Part 1: Concepts and principles (ISO/TC 46/SC 11 ISO 15489-1:2016).* International Organization for Standardization. <https://www.iso.org/standard/62542.html>.

International Press Telecommunications Council. (2005, March 15). *"IPTC Core" Schema for XMP. Version 1.0. Specification document Document. Revision 8.* https://iptc.org/std/Iptc4xmpCore/1.0/specification/Iptc4xmpCore_1.0-spec-XMPSchema_8.pdf.

International Press Telecommunications Council. (2008). *IPTC Photo Metadata Standard 2008. Document Revision 1.* IPTC. https://iptc.org/std/photometadata/2008/specification/IPTC-PhotoMetadata-2008_1err.pdf.

International Press Telecommunications Council, & Newspaper Association of America. (July 2014). *IPTC-NAA*

Information Interchange Model Version 4. Revision 2. <https://iptc.org/std/IIM/4.2/specification/IIMV4.2.pdf>.

IPTC Photo Metadata Working Group. (2020). *IPTC Photo Metadata Standard 2019.1*. IPTC. <https://iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-2019.1.html>.

IPTC Photo Metadata Working Group. (2023). *IPTC Photo Metadata Standard 2022.1*. IPTC. <https://iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-2022.1.html>.

IPTC Photo Metadata Working Group. (2024). *IPTC Photo Metadata Standard 2023.2*. IPTC. <https://iptc.org/std/photometadata/specification/IPTC-PhotoMetadata-2023.2.html>.

ISO/TC 46/SC11. (2008). *N800r1: Where to start. Advice on creating a metadata schema or application profile*. International Organization for Standardization. <https://committee.iso.org/files/live/sites/tc46sc11/files/documents/N800R1%20Where%20to%20start-advice%20on%20creating%20a%20metadata%20schema.pdf>.

JEITA. (2000). *Japan Electronics and Information Technology Industries Association (JEITA) Inaugurated Today*. <https://web.archive.org/web/20211016115710/https://www.jeita.or.jp/english/press/2000/1101>.

Korzybski, A. (1933). *Science and Sanity: An Introduction to Non-Aristotelian Systems and General Semantics*. The International Non-Aristotelian Library Publishing Company.

Metadata Working Group. (2008a, September 24). *Metadata Working Group Introduces First Specification for Interoperability and Preservation of Metadata in Digital Photography [Press release]*. http://www.metadataworkinggroup.com/info/pdf/photokina_pr_2008_09_24.pdf.

Metadata Working Group. (2008b). *Guidelines for handling image metadata: Version 1.0*. Metadata Working Group. <https://web.archive.org/web/20171011044821/http://>

www.metadataworkinggroup.com/pdf/mwg_guidance_v100.pdf.

Metadata Working Group. (2009). *Guidelines for handling image metadata: Version 1.0.1*. Metadata Working Group. https://web.archive.org/web/20180303110558/http://www.metadataworkinggroup.com/pdf/mwg_guidance_v101.pdf.

Metadata Working Group. (2010). *Guidelines for handling image metadata: Version 2.0*. Metadata Working Group. http://www.metadataworkinggroup.com/pdf/mwg_guidance.pdf.

Microsoft. (2021). *People Tagging Overview*. <https://learn.microsoft.com/en-us/windows/win32/wic/-wic-people-tagging#microsoft-photo-12-schema>.

Miles, A., & Bechhofer, S. (2009). *SKOS Simple Knowledge Organization System Reference*. 18 August 2009. W3C. W3C Recommendation. <http://www.w3.org/TR/2009/REC-skos-reference-20090818/>.

Murray, J. D., & VanRyper, W. (1994). *Encyclopedia of graphics file formats*. O'Reilly & Associates.

O'Rourke, J. (1987). *Art gallery theorems and algorithms. The International series of monographs on computer science: Vol. 3*. Oxford University Press.

Pomerantz, J. (2015). *Metadata. The MIT Press essential knowledge series*. The MIT Press.

Preparata, F. P., & Shamos, M. I. (1985). *Computational Geometry: An Introduction. Texts and Monographs in Computer Science*. Springer. <https://doi.org/10.1007/978-1-4612-1098-6>.

Riecks, D. (2005, March 21). *IPTC - IIMv4 & IPTC Core XMP Schema v1.0 Fields mapped to Imaging Program Labels*. https://web.archive.org/web/20050326111959/https://www.controlledvocabulary.com/imagedatabases/iptc_core_mapped.pdf.

Schlegel, J., Carloni, M., Wogrin, S., Graff, A. M., & Verhoeven, G. J. (2023). Making a mark—Towards a graffiti thesaurus. In G. J. Verhoeven, J. Schlegel, B. Wild, S. Wogrin, & M. Carloni (Eds.), *Document | archive | disseminate graffiti-scapes. Proceedings of the goINDIGO2022 international graffiti symposium* (pp. 203–219). Urban Creativity. <https://doi.org/10.48619/indigo.v0i0.710>.

Schneider, P. J., & Eberly, D. H. (2003). *Geometric tools for computer graphics. The Morgan Kaufmann series in computer graphics and geometric modeling*. Morgan Kaufmann Publishers.

Verhoeven, G. J. (2010). It's all about the format – unleashing the power of RAW aerial photography. *International Journal of Remote Sensing*, 31(8), 2009–2042. <https://doi.org/10.1080/01431160902929271>.

Verhoeven, G. J., Wogrin, S., Schlegel, J., Wieser, M., & Wild, B. (2023). Facing a chameleon—How project INDIGO discovers and records new graffiti. In G. J. Verhoeven, J. Schlegel, B. Wild, S. Wogrin, & M. Carloni (Eds.), *Document | archive | disseminate graffiti-scapes. Proceedings of the goINDIGO2022 international graffiti symposium* (pp. 63–85). Urban Creativity. <https://doi.org/10.48619/indigo.v0i0.703>.

Wild, B., Verhoeven, G. J., Wieser, M., Ressler, C., Schlegel, J., Wogrin, S., Otepka-Schremmer, J., & Pfeifer, N. (2022). AUTOGRAF—Automated Orthorectification of GRAffiti Photos. *Heritage*, 5(4), 2987–3009. <https://doi.org/10.3390/heritage5040155>.

Wild, B., Verhoeven, G. J., Wogrin, S., Wieser, M., Ressler, C., Otepka-Schremmer, J., & Pfeifer, N. (2023). Urban creativity meets engineering. Automated graffiti mapping along Vienna's Donaukanal. In G. J. Verhoeven, J. Schlegel, B. Wild, S. Wogrin, & M. Carloni (Eds.), *Document | archive | disseminate graffiti-scapes. Proceedings of the goINDIGO2022 international graffiti symposium* (pp. 131–145). Urban Creativity. <https://doi.org/10.48619/indigo.v0i0.705>.

Yu, L. (Ed.). (2014). *A Developer's Guide to the Semantic Web (Second edition)*. Springer. <https://doi.org/10.1007/978-3-662-43796-4>.